

This electronic thesis or dissertation has been downloaded from the King's Research Portal at <https://kclpure.kcl.ac.uk/portal/>



## **A behaviourist framework for describing open self-organising systems**

Rekkas, Spyridon

*Awarding institution:*  
King's College London

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without proper acknowledgement.

### **END USER LICENCE AGREEMENT**



**Unless another licence is stated on the immediately following page** this work is licensed

under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

licence. <https://creativecommons.org/licenses/by-nc-nd/4.0/>

You are free to copy, distribute and transmit the work

Under the following conditions:

- Attribution: You must attribute the work in the manner specified by the author (but not in any way that suggests that they endorse you or your use of the work).
- Non Commercial: You may not use this work for commercial purposes.
- No Derivative Works - You may not alter, transform, or build upon this work.

Any of these conditions can be waived if you receive permission from the author. Your fair dealings and other rights are in no way affected by the above.

### **Take down policy**

If you believe that this document breaches copyright please contact [librarypure@kcl.ac.uk](mailto:librarypure@kcl.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.

# PHD THESIS

A behaviourist framework for describing open  
self-organising systems

Spyridon Rekkas  
PhD in Computing Science

January 2, 2017

## **Abstract**

When designing large distributed systems, it is common to try and impose an organisational structure, a structure with distinct roles, a defined communication model, and social control aspects. Having an organisational structure allows for observing the system's emerging overall behaviour, as well as controlling it. This seems more necessary when designing open systems where heterogeneous members are allowed to join and leave the system. Open systems especially need an organisational structure that is adaptive to changes, ideally a structure that is self-adaptive without the need of external intervention.

In the case of Multi Agent Systems (MASs), the building blocks of this self-adaptive organisational structure (the roles, communication model and a social order mechanism) need to be defined in an unambiguous way that allows this structure to be applied onto software agents. Most of the existing models for designing Multi Agent Organisations make use of mental notions, such as power, belief, intention, obligation etc. These are notions used in every day language when describing an organisational structure, but can have an ambiguous meaning and semantics. This makes them difficult to implement in a programming environment leaving their actual meaning open to the designer's interpretation and leading to not truly open systems.

This thesis explores the possibility of bridging this gap between organisational design and implementation by defining the main building blocks of an organisational structure using a behaviourist approach. Inspired by the doctrine of Behaviourism, the thesis provides a descriptive framework for specifying an open organisation's building blocks in behavioural terms.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Background . . . . .	5
1.1.1	Characteristics of agents . . . . .	7
1.1.2	Environment . . . . .	7
1.1.3	Taxonomy of agent systems . . . . .	8
1.2	Agent systems of interest to this thesis . . . . .	10
1.2.1	Agent organisations . . . . .	11
1.2.2	Open self-organising systems . . . . .	11
1.3	Agent Oriented Software Engineering (AOSE) . . . . .	12
1.3.1	Framework for MAS . . . . .	13
1.4	Example domain . . . . .	15
1.4.1	Background on Global Equity Markets . . . . .	15
1.4.2	Characteristics of equity trading . . . . .	15
1.4.3	Markets and types of MAS . . . . .	17
1.5	Applications and AOSE approaches . . . . .	19
1.5.1	Prominent AOSE models . . . . .	19
1.5.2	Evaluation . . . . .	21
1.5.3	Summary . . . . .	39
1.6	Open questions . . . . .	40
1.7	Summary of contributions . . . . .	42
<b>2</b>	<b>Organisational Structures and Self Organisation</b>	<b>43</b>
2.1	MASs as organisations . . . . .	43
2.2	Holism . . . . .	46
2.3	Self-organisation . . . . .	47
2.3.1	Reasons for re-organisation . . . . .	47
2.3.2	Reasons for self-organisation . . . . .	48
2.3.3	Approaches to self-organisation . . . . .	49
2.4	Behaviourism . . . . .	52

2.4.1	Notions . . . . .	54
2.5	Organisational model . . . . .	57
<b>3</b>	<b>Roles</b>	<b>58</b>
3.1	Organisation and roles . . . . .	58
3.2	Existing definitions of role . . . . .	59
3.3	Characteristics a role definition needs to have . . . . .	60
3.3.1	Prominent models and Roles . . . . .	62
3.4	Behaviouristic and Mentalistic approaches to roles . . . . .	65
3.5	Definition of role . . . . .	67
3.6	Assignment of role . . . . .	70
3.6.1	When is role re-assignment needed . . . . .	72
3.6.2	How to identify that role re-assignment is needed . . . . .	72
3.6.3	How is role re-assignment performed . . . . .	74
3.7	Alteration of role . . . . .	77
3.8	Summary and evaluation . . . . .	80
<b>4</b>	<b>Communication</b>	<b>82</b>
4.1	Communication in organisations . . . . .	82
4.2	Characteristics of a communication model . . . . .	83
4.2.1	Decoupling . . . . .	83
4.2.2	Signals' creation . . . . .	84
4.2.3	Internal information processing mechanisms . . . . .	84
4.2.4	Communication language . . . . .	85
4.2.5	Communication channels . . . . .	85
4.2.6	Decentralised structure . . . . .	85
4.2.7	Deceitful communication . . . . .	86
4.3	Existing approaches . . . . .	87
4.4	Environment and stigmergic communications . . . . .	91
4.4.1	Environment in MAS . . . . .	92
4.4.2	Environment mediated communications . . . . .	94
4.4.3	Characteristics of environment mediated communication . . . . .	96
4.5	Communications model . . . . .	98
4.5.1	Evaluation . . . . .	99
4.6	Informational content . . . . .	100
4.6.1	Background . . . . .	101
4.6.2	Current definitions . . . . .	101
4.6.3	Skyrms' model . . . . .	105
4.6.4	Expansion of Skyrms' definition . . . . .	108

<b>5</b>	<b>Social Order</b>	<b>115</b>
5.1	Introduction . . . . .	115
5.2	Social order and control in open MAS . . . . .	117
5.2.1	Characteristics of open systems and social order . . . . .	117
5.2.2	Maintaining social order in open systems . . . . .	119
5.3	Social control in AOSE methodologies . . . . .	122
5.4	Social order in framework . . . . .	123
5.4.1	Evaluation . . . . .	127
<b>6</b>	<b>Framework</b>	<b>129</b>
6.1	Complete framework . . . . .	129
6.1.1	Roles . . . . .	129
6.1.2	Communication . . . . .	131
6.1.3	Social order . . . . .	131
6.1.4	Framework steps . . . . .	132
6.1.5	Diagrammatic representation . . . . .	134
6.2	Use in example domain . . . . .	136
<b>7</b>	<b>Summary</b>	<b>146</b>
7.1	Questions answered and contribution . . . . .	146
7.2	Open questions for future work . . . . .	147

## Acknowledgements

From old books, to random ideas, discussions, evenings by the sea, research, observing people, the workplace, all contribute to coming up with questions and heading on to answer them. Many people helped me through this process from random ideas to this dissertation, and I would like to thank them.

Firstly, my supervisor for many years, Emeritus Professor Andrew Jones, who was inspiring, captivating, extremely helpful and guided my mind through a labyrinth of thoughts to manage to think in different ways. My supervisor the last year, Professor Peter McBurney who guided me through the process of putting my research on paper, providing important and very useful guidance. Also, my second supervisor, Professor Michael Luck, who, with his feedback and to the point questions, helped me think in a more cohesive way and showed me how I should approach my thesis.

I would not have studied for this PhD without the financial support of King's College London and I would not have continued without the support of my employers these years.

On a personal level, family and friends have all supported me during these years in any possible way. Firstly and foremost, I am indebted to my parents, Michail and Anna, for giving me the mindset to question things and for their continuous support. My brother Dimitris holds a special place in this for our endless discussions. Finally, all my friends, some being in the same endeavour as myself and offering guidance and long discussions, some for discussing subjects from psychology to information theory, others accepting that 'I am weird', and everyone for their invaluable support.

# Chapter 1

## Introduction

### 1.1 Background

During the second half of the last century the emergence and fruitful exploration of the concept and practice of artificial intelligence, together with advances in cognitive sciences have led to the creation of complex and intelligent computer applications, which aimed to represent in their functions an analogy to human cognition. However, over the last few years there has been a growing interest towards creating artificial societies using computer programs which are autonomous, have a certain ability to reason and to communicate but are not trying to completely mimic human reasoning and behaviour. Their aim is, instead, to successfully perform tasks utilising their social structure. These programs are called agents and the artificial societies designed, which in the general form contain both human and computer agents (Dignum et al., 2002a), are termed Multi Agent Systems (MASs).

The term of agency is common in every day life; an estate agent is tasked with finding appropriate property and negotiating on one's behalf, or a job agent acts as an intermediary between the employer and potential employees. In general, an agent acts as a delegate for performing a certain task. Why are agents needed? Simply to take over time consuming tasks, or because of their advanced knowledge and skills in specific jobs. A software agent is used for the same reasons.

When more than one agents are involved in a task, then they participate in a multi agent system. The house searching example is one where multiple agents can participate. The potential buyer is talking to several agents, depending on the property they are interested in, and these agents might be negotiating



with other agents, and so on. In a similar manner, software based multi agent systems can be formed.

MASs are used in different industries and various business domains, from production lines and email filtering applications to electronic procurement and trading platforms. With regards to commercial applications of agent systems, a recent survey by (Müller and Fischer, 2014) can provide a good insight of the different sectors that MASs have been applied onto, as well as the level of maturity of these applications. Intelligent agent systems and Multi agent systems have been designed and developed mainly in academic environments, whilst a few applications having been developed as joint ventures between academia and industry, and others by the industry alone.

To gather a list of candidate systems, the authors issued an open call to academia and the industry requesting information on candidate systems to be assessed as per their maturity. They also pursued contacts from the AAMAS (Autonomous Agents and Multi Agent Systems) conferences, and they personally contacted major figures of the sector. After getting this list, the authors contacted the developers/designers of the nominated applications requesting more information on the level of use, the background of the application, and the impact on particular sectors among others. Most of the industrially deployed applications aid Logistics and Manufacturing, the Aerospace, and Energy sectors. These systems are used for control, scheduling, production planning, and material handling in the Manufacturing and Logistics sector (Leitão and Vrba, 2011). In the Aerospace and Defence industry, agent systems have been deployed for control, e.g. for controlling unmanned vehicles, and simulations. Simulations and forecasting energy demands are also the main uses of agent systems in the Energy sector.

Agent systems are also in use in other business applications, such as e-commerce, electronic auctions (Müller and Fischer, 2014), as well as in tourism services and games (Sturm and Shehory, 2014).

Multi Agent Systems introduce a different view in developing technology solutions. They represent a collaborative approach, while traditional computer systems are developed using a hierarchical command and control approach. This change can be mapped to the evolution of business operations during the past years, which have moved away from a top-down, silo, approach towards a flatter, collaborative, approach. As a result, enterprise computer systems that support operations need to be flexible and decentralised. MASs can offer this flexibility as, in some implementations, their behaviour emerges from local components' interaction instead of being mandated from a centralised authority (Odell, 2007a).

### 1.1.1 Characteristics of agents

There are slightly different views of what an Agent application is among researchers Sturm and Shehory (2014). However, most definitions agree that an agent has the following characteristics:

1. Autonomy; agents can reason for themselves and can act without waiting to be invoked or to be explicitly directed what to do.
2. Adaptability; agents can adapt to changes in their environment.
3. Social ability; agents can communicate with other entities and their environment.

Odell suggests that there are three commonly accepted properties that characterise an agent: autonomy, interaction and adaptivity (Odell, 2007a). Firstly, a software agent has the ability to reason about its goals and the state of its environment and is capable of decision making. Interaction means that an agent is not seen as a stand alone entity, but communicates with other agents and interacts with the environment to achieve its own or the system's goals. Lastly, an agent is not static; it can adapt its behaviour based on its reasoning and the interaction with the environment and other agents.(Dignum and Dignum, 2006), (Odell, 2007a).

According to Wooldridge and Jennings, an agent has the following properties: autonomy, reactivity, pro-activeness and social ability (Wooldridge and Jennings, 1995). An agent has internal states, that are not visible to other agents, and is capable of decision making based on these states. In addition, agents interact with their environment and react to changes in it adapting their behaviour; this is the characteristic of reactivity. On the other hand, agents are proactive as they have internal goals that they aim to achieve. Finally, agents have the characteristic of social ability. They interact with other agents and engage in social activities in order to achieve their goals.

Luck et al. (Luck et al., 2005) define an agent as a computer system, which can act autonomously in a dynamic domain where, typically, multiple agents operate. This definition suggests the same characteristics as the definitions above: autonomy, adaptability and social ability.

### 1.1.2 Environment

As Odell et al. say, “without an environment, an agent is effectively useless” (Odell et al., 2003b). The environment is where the agents exist and operate. From a general viewpoint, any object that is not a member of the system can

be considered as its environment. Without an environment, agents would not be able to function.

Moreover, the system's environment is not a passive entity. It has its own functions, it changes overtime and can affect the system's behaviour. The environment itself is active, has its own processes and can change its state, independently of the actions of the agents (Parunak, 1997).

### 1.1.3 Taxonomy of agent systems

Agent systems have been applied onto different sectors, with implementations varying from proof-of-concept systems to mature industrial level systems. Given that agent systems are used to meet different business needs, from planning and scheduling, to control and simulations, one understands that there are different types of agent systems.

(Müller and Fischer, 2014) follow the classification below for agent systems:

- Multi Agent Systems

Multi Agent Systems (MASs) focus on coordination, and cooperation between agent applications. MASs are used for tasks that require collaboration, and need a group of agents to work together in order to achieve a common greater goal. Control systems for production lines are a good example of MASs.

- Intelligent Agent Systems

Intelligent Agents focus on aspects of a single agent, such as planning or learning. For example, an agent which is tasked with controlling the temperature of a room is considered to be a planning agent. Using its sensory input, it can make an informed decision and plan on the next action; raise/keep/lower the temperature.

- Personal/UI Agent Systems

Personal agent systems focus more on the agent-human interaction. A simple form of a personal agent is an email filtering agent, tasked to filter its user's emails into folders and learning from the user's behaviour.

Their survey shows that the majority of deployed systems can be classified as Multi Agent Systems (82%), followed by a similar percentage of Intelligent Agents (9%) and Personal/User Interface agents (7%). It is evident that more focus has been given to MASs.

One could continue the taxonomy of MASs, into the following categories De Jardin (2014)

- Cooperative

MASs designed to achieve a common goal through the cooperation of situated or mobile agents. Example applications of cooperative agents are: Distributed sensor networks, Distributed vehicle monitoring, Distributed delivery (logistics), and DAI (Distributed Artificial Intelligence) systems which divide a problem into smaller tasks and distribute their solution between different members of the system.

- Competitive

MASs which do have a common goal, but the behaviour of their members is not expected to be cooperative. The system's goal is achieved by the members competing with each other; for example for resources. Some example applications of competitive MASs are simulators of energy consumption, or of marketplaces.

Competitive MASs can pose a few issues to their designers and administrators, such as how to achieve fairness, stability, and how to deal with deceptive members. Researchers have suggested and implemented various techniques to deal with these issues, such as Voting mechanisms, Auctions, or Contract nets.

The taxonomy of MASs could also follow a different path, by describing MASs based on the design of their members. If all members of the system have been designed by the same group, or under the same specifications, we can call the system Homogeneous. While, if some members of the system have been designed and implemented by different groups, the MAS is called Heterogeneous.

Heterogeneous systems are applied on Open environments where it is not possible, or desirable, to control the implementation of all the members. Some examples of open systems are the following:

- Scalable systems where agents developed by different groups need to participate, due to the large number of potential participants.
- Auction systems where not all participants are known. Electronic procurement systems for example, cannot presume that every participant has been developed by the same group and under the same specifications. It is the competitive nature of the system that calls for heterogeneous participants.

## 1.2 Agent systems of interest to this thesis

This section introduces the specific type of agent systems that are of interest in this thesis, based on the research and the taxonomies above. The thesis deals with heterogeneous, competitive, open, self-organising Multi Agent Systems:

### 1. Multi Agent Systems

As seen in the analysis of existing agent applications, MASs form the majority of applications developed. MASs are seen as a reflection of the collaborative business world.

### 2. Heterogeneous

In a MAS different groups need to work together to achieve a greater goal, meaning that it is hard to expect and achieve homogeneity in large MASs. In addition, allowing for heterogeneity, makes the system more scalable.

### 3. Competitive

As implied by the example domain used throughout the thesis and detailed in section 1.4 below, my applied interests lie on electronic markets which are by definition competitive. The past years there have been advancements in aspects surrounding electronic trading, from market data analysis, to trading algorithms, and post-trade analytics. However, the perception of markets individually and as a network of connected nodes has not been explored much by industry. Competitiveness goes hand in hand with deceit. As in real life organisations, it is difficult to think of an organisation with competing members which are always trusted and behaving to the norms.

### 4. Open

By open, one means systems where their members can join or leave during runtime. This means that interactions between members can be forced to change D’Inverno et al. (2012) and that functions performed by certain members need to be performed by other members. Openness is usually a characteristic of heterogeneous, large systems. The thesis is interested in truly open systems where agents’ architecture is left unspecified Ferber et al. (2003).

### 5. Self-organising

Open systems are volatile, since one cannot predict which and how many members will be participating at each given time. The requirement for

scalable systems, along with the limited resources/time that an administrator could have, call for self-organising structures that can adapt to changes in their environment.

### 1.2.1 Agent organisations

The nature of MAS applications and that of the agents themselves has driven research to use paradigms from human organisations for the analysis and specification of MASs (for a review, see (Coutinho et al., 2005)). These organisations drive and guide how their members behave and interact with each other over the course of the system’s runtime. An organisational structure is not always explicitly imposed on MASs. However, all MASs contain roles, relationships and structures of authority, even in an informal way (Hörning and Lesser, 2005). Despite the fact that a widely accepted definition of what an organisation is does not exist, as different sciences provide different definitions, several researchers agree that there are four attributes that can characterise an organisation:

1. The system’s high-level purpose;
2. The roles that exist in it;
3. The methods of communication and
4. a function to evaluate its performance, a way to ensure that it is doing what it is designed to do.

(see for example (Sims et al., 2003) and (Parunak and Odell, 2001)).

Moreover, there is a tendency to specify MASs as self-organising systems that do not have a static organisational structure. As MASs are used for large scale, complex systems, researchers have argued that the organisation needs to be flexible and able to adapt its structure when needed (for example see (Di Marzo Serugendo et al., 2006)).

### 1.2.2 Open self-organising systems

A system being open, means its observers cannot know the internal structure of all the participants. They cannot know how these participants reason, or if they are able to reason, in different situations. As a result, one cannot predict their behaviour with certainty. It is possible that members of the system will not behave in an expected, or prescribed way. Also, it is possible that they will not share the common goal of the system, that they might try to take advantage

of or halt the system, or simply that they are not fit for purpose. (Artikis et al., 2009)

The openness of a system can lead to situations where any organisational structure imposed on the system, any predefined behaviours and interactions, would need to be realigned. A self-organising system is one that can adapt its organisational structure, without external interception (Schmeck et al., 2010). Open self-organising systems are the focus of this thesis as they provide interesting characteristics, e.g. emergent properties and resilience, and can be applied to today's collaborative electronic world.

### 1.3 Agent Oriented Software Engineering (AOSE)

Agents represent a different way of designing and building software programs, and are good candidates for specific applications due to their characteristics. As with any type of application, the different research paths followed in the agent community, along with the different types of agent applications, have led to various software methodologies being developed.

Agent Oriented Software Engineering (AOSE) deals with the analysis, design, and implementation of software systems related to agency. (Sturm and Shehory, 2014) argue that there are two ways of viewing AOSE. The first is related to efforts in supporting the development of agent-based systems. The second way aims to support the development of complex systems where the notion of agency is being used. The authors state that "while it appears that the AOSE community made considerable progress towards the first, the second has not advanced much". This is the same distinction made by (Luck et al., 2004); the authors distinguish between aspects of AOSE targeted at analysing, designing, and building agent systems and aspects aimed at designing large complex systems using agents.

(Sturm and Shehory, 2014) provide a thematic map of Agent Oriented Software Engineering (AOSE). They break down AOSE into the following categories:

- Agent Oriented Methodologies
  - Agent oriented modelling techniques which deal with ways of modelling agents and how they operate in their environment.
  - Agents and model-driven approaches.
  - CASE (Computer Aided Software Engineering) tools to support agent-oriented software development.

- Evaluation and comparison of modelling techniques and methodologies.
- Agent Oriented Frameworks
  - The required building blocks of an agent-oriented framework.
  - Relationship between research driven frameworks and industry generated frameworks.
- Agent Programming Languages
  - Different types of programming languages and their applications.
  - Logics for agent-based programming.
  - Benchmarks and Verification techniques of agent programming languages.
- Agent communication
  - Content Languages.
  - Interaction Mechanisms.
- Agent Oriented Architectures
  - Software architectures for MAS.
  - The desired properties for such architectures.
  - Reuse of agent based systems design knowledge, patterns, and reference architectures.

### 1.3.1 Framework for MAS

This thesis' aim is to provide a descriptive framework for designing the type of MASs discussed above (open, competitive, heterogeneous, and self-organising), using a behaviourist approach. The framework falls into the Agent Oriented Methodologies theme, with the note that it is dealing with Sturm and Shehory's second interpretation of AOSE; the design of complex systems where agents play a central role. The framework is aimed to be the basis for creating an AOSE for agent based organisations, rather than for agent-centric applications. Moreover, it focuses on such organisations that can self-adapt their structure using a bottom-up approach.

The task to produce a full methodology is too large for a PhD thesis. In the descriptive framework, the following will be covered:



1. A discussion of the underlying concepts and principles, the blocks, needed to describe the organisational structure of such a system.
2. A description of each block in more detail and of how these concepts tie in together.
3. A description of what type of systems are to be designed using the framework.
4. This description will be done using a notation.
5. A high level process describing how the building blocks can be defined, using the notation, for designing a specific system.

(Dam and Winikoff, 2013) view a methodology as including the following:

1. A notation: a way of describing the specified system, in a clear unambiguous way.
2. The underlying concepts: a definition of the concepts that form the methodology. For example, what is meant by Belief/Desire/Intention in a BDI methodology.
3. An indication of which models are to be developed: since not all methodologies can, or should, be applied to all types of models.
4. A process: that defines the steps a designer can produce a specification of a specific model, using the notation prescribed and in alignment to the underlying concepts.
5. Detailed guidelines on each level of the process: to aid the designer in the process defined.
6. Tool support (highly desirable but not necessary): specific tools that the designer could use to produce a specification, just like TeXworks is a tool for writing tex documents.

This thesis, proposing a framework and not a methodology, does not provide detailed guidelines, or tool support for developing the specific type of MASs. These two items aim to provide details steps and support for organisations to adapt a framework into their needs, build a methodology, and processes that support this methodology.

## 1.4 Example domain

The example domain that will be used to better understand and provide real life situations when discussing this thesis is the regulated equity trading markets. This refers to the secondary trading places of equity titles, such as the London Stock Exchange or the NASDAQ OMX markets. This sector has evolved to be a global, fairly open, system where different members interact and compete with each other, each one with a common goal; to maximise their profits.

### 1.4.1 Background on Global Equity Markets

An equity market is a place where public equity shares of companies are traded. Most companies need to receive external funding in order to grow. This is acquired through various means such as loans, private funding, or by selling equity stocks; by selling parts of the company to private investors or to the public. It is the public stocks of a company that are traded in an equities market.

The company needs to get evaluated and, when a price is agreed, a share of the company is offered to the public, in the form of an equity stock. This is the primary market.

These stocks can then be transferred, bought and sold, as long as the company remains public. The negotiations, agreements, and transactions of transferring stock are regulated and are performed on a secondary equity market place, such as the London Stock Exchange.

### 1.4.2 Characteristics of equity trading

One can broadly identify four different types of members in equity markets:

- A Fund, which is the institution that holds large amounts of money and are willing to invest in equity in order to maximise their return on investment. The funds do not have direct access to the markets. Their purpose is to collect investments, e.g. pension funds, and return a profit to their clients when agreed.

The funds sell their services to their clients, and their selling point is that they know which stock to buy or sell. The funds are clients of the brokers, and are also referred to as the buy-side.

- A Broker, which has direct access to trading on the market and either receives orders from the fund or trades for themselves.

The brokers, also called sell-sides, sell their services to the buy-sides, and their selling point is that they know how to buy and sell a stock.

- A Clearing house, which is an intermediary between the brokers and the funds and takes care of the logistics, e.g. of confirming the trades, of settling differences etc...
- The Market itself, i.e. the stock exchange, which is a regulated place where brokers can trade stock. There are hundreds of markets around the world, each one operating in a similar way.

A market has defined opening hours, and defined periods of continuous trading and auctions. Continuous trading means that every participant places their buy (bid) and sell (offer) orders at a specific price and volume. When two prices match, then a trade is agreed. An auction is announced at the start and end of continuous trading sessions and intra-day to regulate the price if needed. An auction accepts bids and offers over a, random, period of time and prices are crossed at the end of the auction.

The secondary equities trading markets domain has the following characteristics:

- The aim of each member is to maximise their profit. This can occur either by trading equity stocks, selling at a higher price than bought, or by providing a service to other members.

Funds maximise their profit by carefully building up their portfolio of stocks. Their aim is to earn profits by buying and selling stock at adequate times so that they have sold at a price higher than the one they paid to buy the stock.

Brokers make profit by offering their trading capabilities and knowledge to funds. Brokers compete with each other in order to get more clients, funds, and on the market in order to get a better price for their orders.

Clearing houses maximise their profit by confirming and settling trades between brokers and the market. Whilst clearing houses had a monopoly until recently, with one house operating in each country, they are now in a competing environment where they can offer their services to multiple markets.

Exchanges make profit by charging brokers for entering orders and trading on them. With multiple stocks being traded across many exchanges, even globally, exchanges compete with each other in order to get liquidity; bid and offer orders from brokers.

- The market is, at the majority of times, in equilibrium. This equilibrium emerges from the fact that the goals of each member are conflicting. One needs to have a negative profit for another member to see a gain.

### 1.4.3 Markets and types of MAS

Equity trading markets are a good example of an open, heterogeneous, competitive, self-organising system. Assessing a market against the characteristics of such a system as described above:

- Members of the global equities market can join and leave on demand, with some of their actions being regulated by local regulative bodies, but without central control. In addition, despite the global nature of the system, each member has a detailed view only of their local environment, be it exchange, country, or continent, due to technological and time constraints.

Here, one needs to make a distinction between two different cases of openness in this example domain. Firstly, at a global level, firms of any of the four roles described above can join or leave the system. This is a slow process which is regulated at a local level, but is open in the sense that its participants can join and leave at will. Secondly, at an Exchange level where brokers join and leave the trading of a specific stock numerous times during day, based on their preference and the current market conditions.

- The system is heterogeneous and one cannot know the internal structure of its members.

Each participant's structure is not known to other participants. The way a company operates, their decision making process, or their long term goals are not made visible to other companies. On the contrary, companies try to withhold information from other participants so as to not influence the market prices against them. For example, if a fund wants to make a large trade, they can try to hide it by ordering multiple smaller trades, try to reach an agreement with another participant outside the market or even try to influence the price in their favour before performing the trade. The way this decision is made, the strategy of the trader or even how this strategy is realised, are not known to their counterparts. The other participants can only observe the firm's behaviour on the market.

- The members of the system do not necessarily share a common, global goal; they are competing against each other.

A regulated market is an example where the participants do not share a common goal. At a high level it can be said that they share the goal of keeping the market running. It is in their interest that the market continues to exist. It is also in their interest that liquidity continues to exist, that when they want to trade on a stock, a participant will be there to fulfil this request.

However, on a day to day basis it can be said that the market participants (especially the funds and the brokers) have conflicting goals. The seller of a stock wants to achieve the highest possible price while the buyer wants to pay the less amount possible.

- Their behaviour cannot be predicted a priori.

The behaviour of members and of the market itself cannot be predicted in advance. Sellers and buyers will act to their own interest in order to get the best price possible. However, one cannot know their strategies for doing so. In regulated markets a great deal of analysis is performed in order to try and predict participants' behaviour; both qualitative and technical analysis. Qualitative analysis is trying to predict the members' behaviour based on information, on news, and trying to predict the general sentiment. Technical analysis is trying to predict the future behaviour based on past trades; trying to reveal patterns of behaviour and project them to future scenarios. There are numerous examples where this analysis is not successful, e.g. the price of a stock moving to the opposite direction, business news negating the results of analysis and so on.

- Finally, the secondary regulated markets can also be seen as a self-organising system based on the way self-organisation is defined in this thesis. The system itself is always trying to alter its structure in order to continue functioning, without external intervention. This can be seen both at a micro level where, for example, if a member changes their trading behaviour, the rest of the members will react to this change keeping the market in equilibrium. It can also be seen at a macro level. The change in the environment, seen as changes in regulation or changes in the market structure itself, does not have an adverse effect on the market itself. Companies alter their operations, take up different business ventures and the market continues functioning.

The example domain of the secondary equity trading markets, both at a macro and micro level, will be used throughout this thesis to discuss existing

AOSE methodologies, explain the building blocks of the suggested framework, and be the example to demonstrate the suggested framework.

## 1.5 Applications and AOSE approaches

The aim of this thesis is to provide a descriptive framework for designing open, heterogeneous, competitive, self-organising MASs. The following section reviews existing prominent AOSE approaches, provides the evaluation criteria for the systems the thesis is interested in, and evaluates them against these criteria.

### 1.5.1 Prominent AOSE models

A large amount of Agent Oriented Software Engineering (AOSE) models have been presented during the past 30 years, with focus on different aspects of agent systems, and on various application domains, each methodology having its unique strengths and weaknesses. In order to identify mature, recent, and active AOSE models with a focus on open systems, three publications have been identified that provide surveys from three different viewpoints.

Dam and Winikoff (2013) perform a structural comparison of existing and active AOSE models based on documentation, maturity, and tool support. The authors' aim is to identify AOSE models that can be used for developing real life applications, review their characteristics, commonalities and differences, and suggest a unified approach. They have identified several AOSE models that have sufficient documentation to be used by researchers other than their creators. These models have also been and are used by the research community with a number of implementations based on them, and with sufficient tool support to build actual systems based on them.

Müller and Fischer (2014) are interested in assessing the adoption of MAS technology by the industry. They issued an open call to academia and the industry for nominating applications and models. Following this, they sent a survey to prominent members of the sector and to the contacts of the nominated platforms. Whilst the authors are interested in analysing the penetration of MAS technology to the industry, their survey provided a comprehensive list of AOSE models used for real life applications. This is important as for an AOSE to be used by the industry, it means that it is mature enough, it has good documentation, and is fit for purpose for real-life applications. It is also important to note that, according to the authors' survey, a third of the applications that data was made available for did not use an existing model.

(Müller and Fischer, 2014) have also provided analysis with regards to the adoption of MASs in different industry sectors. This showed that the most active sector in implementing MASs is Logistics and Manufacturing. This is where Leitão and Vrba (2011) focus on; the implementation of MASs for decentralised control and industrial automation. The authors thoroughly researched relevant publications and company websites to identify real life applications where MASs have been used for decentralised control and automation. Based on their research, they also identify three AOSE models that have been used.

The three references above provide a list of AOSE methodologies as below:

1. JADE and its extension WADE.
2. Tropos
3. O-MaSE
4. Gaia and Extended Gaia
5. Prometheus
6. INGENIAS
7. PASSI
8. PROSA

The list above provides a good and comprehensive list of AOSEs to review and evaluate for their ability to design MASs of interest to this thesis. Reviewing the list, one can see that the following well known models are missing, and hence, need to be included as well:

9. ADELFE
10. Electronic Institutions
11. OperA
12. Moise
13. Communicating open systems

It needs to be noted that models identified by Müller and Fischer (2014) that are proprietary models with no or a small number of academic publications, models that don't appear to be actively used anymore, and models that documentation and examples cannot be acquired, have been excluded. These are the following:

- **KOWLAN.** KOWLAN is a MAS platform designed for diagnosis on telecommunication networks. KOWLAN is a proprietary model of Telefónica España, S.A., Spain and documentation for the processes could not be found.
- **JACK and CoJACK.** JACK, and its extension CoJACK, are excluded from this analysis as they are now propriety of the AOS group. Also, it is not clear, from the existing publications, how they are used for Multi Agent Systems as they are more focused towards cognitive BDI agents.
- **Living Systems Technology Suite (LSTS).** LSTS was identified as a model used for real-life applications, however there is no documentation publicly available on the suite as it is a proprietary suite of Whitestein Technologies Whitestein Technologies (2016).
- **ADEM** which is a framework rather than a methodology, and was suggested as the framework for designing MASs using AML Abushark et al. (2016). Also, (Dam and Winikoff, 2013) suggest that ADEM does not seem to be active anymore.

### 1.5.2 Evaluation

As the aim of the thesis is to provide a descriptive framework for designing open, heterogeneous, competitive, self-organising systems, criteria for evaluating any existing and suggested models need to be defined.

1. **Organisational structure for MASs.** The model needs to be based on an organisational view of the system, to have a way of defining an organisational structure, and of maintaining it during runtime. This is an important criterion as it is claimed that there is a gap between the analysis and development phases of AOSE methods. One of the reasons for this gap being the fact that some platforms do not deal with the organisational aspect as a first class entity Uez and Hübner (2014).
2. **Heterogeneous.** The model needs to allow for members designed and developed by different teams to join and leave the system. It also needs to provide a mechanism, a way, for this to happen.
3. **Competitive and Deceitful.** Another characteristic that the model needs to have is to allow for competitive and deceitful agents to be part of the system, and have a mechanism of dealing with them.



4. Allow and help the **resulting system to be open**. This means that the system needs to allow for agents to join and leave the system, without the system's overall goal being jeopardised. For a system to be open, it needs to be independent from specific architectures and tools. This is to say that the agents' internal architecture and the actual implementation platform should not play a role during the high-level design part of the process Steghöfer et al. (2014).
5. Have mechanisms for allowing the system to self adapt its organisational structure during runtime, in order to be able to react in changes if needed. Ideally, this **self-organisation** goes into two levels:
  - Re-assignment of roles. Roles need to be decoupled by specific agents. Also, there needs to be a mechanism for orphan roles to be assigned, or claimed by agents.
  - Evolution of roles. As in real-life organisations, roles can evolve overtime. A methodology for designing open self-organising systems needs to cater for this evolution of roles when it is needed. As Bernon et al. (2002) state, a self-adaptive software needs to evaluate its own behaviour and change it when this evaluation indicates that it is not achieving what it is intending to do, or if there is a better way of achieving it. In an open, dynamic system, it is important for the system to be able to adapt to changes. Since its behaviour is defined by the roles of the institution, it is important to allow for the roles themselves to change, to evolve based on changes in the environment and based on new capabilities brought to the system by new agents.

The sections below review the most prominent models as identified in 1.5.1 above, and review them against the five evaluation criteria.

## JADE

JADE (Java Agent DEvelopment Framework) Nikraz et al. (2006) is a platform developed by the Telecom Italia Lab (TILAB) in Italy.

The analysis phase of the JADE methodology consists of the following steps:

1. Produce Use Cases
2. Define Agent Types
3. Define responsibilities for each Agent Type (similar to defining roles in other methodologies).

4. Acquaintances Identification where relationships between agent types are identified.
5. Agent refinement where decisions are made as to whether extra agents are needed for supporting the system, how agents identify themselves, and if any supporting information is needed for some agent types.
6. Agent Deployment Information

The next phase for the JADE methodology is the design phase, which is linked closely to the JADE platform. The design step contains steps for re-specifying the agent types if needed, defining the interaction patterns between agents and the way they interact with their environment, the way they register themselves to the system, as well as their internal behaviours. Furthermore, this step models in detail the ontology of the system; concepts, predicates, and agent actions.

Evaluating the methodology against the criteria set:

1. **Organisational structure for MAS.** Yes, especially through the WADE extension, which provides the administration and fault management modules.
2. **Heterogeneity.** Partly. The framework is FIPA compliant, and also the resulting platform can be distributed across multiple machines. However, agents need to be based on the JADE platform.
3. **Competitive and deceitful agents.** Partly. The methodology does not deal with competitive and deceitful agents directly. It is possible to accommodate these types of agents through the WADE control mechanism, but it is not clear as to how this can be achieved.
4. **Open system.** Partly, as agents need to be based on the JADE platform and follow the exact communication patterns specified during the design phase. However, in the specific scenario of Nikraz et al. (2006) where a mobile cinema organiser system is developed, the system can be considered as open since different JADE mobile agents can join or leave the system.
5. **Self-organisation** No, for both aspects below. As per Nikraz et al. (2006), "the organizational structure of the system is static, meaning that non-emergent behaviour at runtime is not expected, and thus, not considered".
  - (a) **Reassignment of roles.**
  - (b) **Evolution of roles.**

## Tropos

Tropos focuses on providing the following with regards to AOSE: a software development process, and formal tools and methods through its extension 'formal Tropos'.

The Tropos methodology has the following four steps:

1. Early requirements
2. Late requirements
3. Architectural design
4. Detailed design

These four steps utilise three different models, and the detailed design is produced through an iterative process of going through these models. Firstly, the Actor Model represents what we would call a role. It consists of the Goal Model, the Plan Model, and the Capability Model which defines the contributions of each top level goal. Secondly, the Dependency Model which defines the relationships between actors and draws the organisational structure of the system. Finally, the Mixed model which combines the two and is used to refine the actor and dependency models through the process. It is important to note that Tropos does not utilise the notion of agent during the analysis phase, it uses the abstract notion of actor allowing for decoupling the analysis and the end implementation.

Evaluating the methodology against the criteria set:

1. **Organisational structure for MAS.** Yes. Tropos is more agent focused Wooldridge (2009), but its dependency model is responsible for representing the social structure of the system.
2. **Heterogeneity.** No, as TROPOS is focused on designing closed systems where the designer has control over the agents that enter the system Dastani et al. (2004).
3. **Competitive and deceitful agents.** No as during runtime there is no mechanism of ensuring that the members follow the specification of the system Arcos et al. (2005).
4. **Open system.** No, as TROPOS is designed for closed systems.
5. **Self-organisation**

- (a) **Reassignment of roles.** Partly, as the architecture is tasked with maintaining the plan during runtime.
- (b) **Evolution of roles.** No, TROPOS does not cater for evolving roles during runtime.

## O-MaSE

Organisation Multiagent Software Engineering (O-MaSE) is a methodology framework and its goal is to allow designers to create customised agent oriented software development processes. DeLoach and Garcia-Ojeda (2014). O-MaSE is based on three basic structures: a metamodel, a set of methods fragments, and a set of guidelines. The metamodel defines the concepts needed when designing and implementing multiagent systems. The method fragments are tasks that need to be done in order to produce specific design outcomes; e.g. models, documentation, or code. Finally, the guidelines show how method fragments are related to each other, for example which outcomes are a prerequisite for other fragments. O-MaSE also provides a set of tools to be used during the analysis and design process.

O-MaSE is different to the other methodologies as it does not define a specific workflow or phases to be followed when building a system. It provides tasks and activities that need to be completed, allowing the designer to decide how and when these are performed, as long as they adhere to the guidelines.

The O-MaSE metamodel views MASs from an organisational perspective and defines the following concepts:

- Goal, which defines objectives of the organisation.
- Role, which captures behaviours to achieve specific goals.
- Agent, that plays roles within an organisation.
- Organisational Agent, which is a group of agents acting as one agent in this organisation. This is the same abstraction as the holonic agents suggested by Schillo et al. (2002).
- Capability, which denotes the abilities of an agent.
- Domain model, that describes the environment within which the organisation runs.
- Policy. Policies constrain the behaviour of the organisation and they are often expressed as liveness and safety properties, similar to the Gaia model Wooldridge (2009), p.187.

- Protocols that define how agents, roles, or actors of the environment interact with each other.
- Actors that are entities of the environment and interact with the system.
- Plan. A plan describes the agents' planning algorithms.

Evaluating the methodology against the criteria set:

1. **Organisational structure for MAS.** Yes, as the O-MaSE metamodel is derived from the Organisation Model for Adaptive Computational Systems (OMACS) DeLoach and Garcia-Ojeda (2014).
2. **Heterogeneity.** No. The model requires the designer to describe the internal algorithms of the agents, as part of the 'plan' concept.
3. **Competitive and deceitful agents.** No, O-MaSE contains the design of agents as part of the methodology, and does not have a mechanism for dealing with deceitful agents as such.
4. **Open system.** Yes, the O-MaSE metamodel supports the development of open systems Garcia et al. (2015).
5. **Self-organisation**
  - (a) **Reassignment of roles.** Yes, as "it provides suitable mechanisms for allowing the system to reorganise at runtime in order to adapt to its environment and changing capabilities" Garcia et al. (2009).
  - (b) **Evolution of roles.** No, O-MaSE does not cater for redefining roles during runtime.

## Gaia

The Gaia model (Zambonelli et al., 2003), utilises two types of concepts; abstract that are used during the analysis phase but are not necessarily represented in the end system, and concrete that are representations of items built in the system.

In Gaia, the end system is seen as an organisation which is realised by roles, which have permissions and responsibilities dictating their behaviour against other roles, and activities that are internal actions for each role. Responsibilities are divided into liveness properties, i.e. responsibilities to do something good, and safety properties, i.e. responsibilities to prevent something bad. The communication between roles is defined by protocols.

During the design phase, roles are assigned to specific agents, and are interpreted in terms of permissions, responsibilities, activities and interaction with

other roles. The environment is also defined in the model, with the available and allowed interactions between it and the agents specified at the design phase. Similar to Tropos, Gaia allows for decoupling by utilising the abstract notion of role in the analysis and assigning roles to agents at the design phase.

The extended Gaia methodology focuses more on the organisational aspects of a system, following the steps below for analysis and design:

- Analysis
  - Decomposition of the organization in sub-organizations
  - Environment Model
  - Preliminary Roles Model
  - Preliminary Interaction Model
  - Rules of Organization that are divided into liveness and safety rules
- Design
  - Organizational structure: Topology and Control structure
  - Final Roles Model
  - Final Interaction Model
  - Agent Model
  - Service Model

Evaluating the methodology against the criteria set:

1. **Organisational structure for MAS.** Yes, as the methodology guides developers into viewing agent based systems as an organisational process design Wooldridge (2009).
2. **Heterogeneity.** Yes. Roles are decoupled from actual agents, and Gaia does not predefine the internal architecture of agents.
3. **Competitive and deceitful agents.** Yes, through the safety rules defined during the analysis phase.
4. **Open system** No. The methodology does not support open agent systems, because it does not treat agents as given entities Dastani et al. (2004).
5. **Self-organisation** No, as Gaia does not provide re-organising mechanisms during runtime, neither for re-assigning nor for re-designing roles.
  - (a) **Reassignment of roles.**
  - (b) **Evolution of roles.**

## Prometheus

The Prometheus methodology Padgham and Winikoff (2002) is an iterative methodology focused on designing intelligent BDI agents, and consists of three stages Wooldridge (2009):

1. System specification. This step identifies the goals of the system and its basic functionality. It also defines the interface between the system and its environment in terms of inputs, or percepts, outputs, or actions, and external data.
2. Architectural design which identifies the agent types, the patterns of interaction, and the system's structure. At this stage the designer groups functionalities identified on the first step into agent types through an iterative process.
3. Detailed design which focuses on the internal design of the agents and refines them into their capabilities.

Prometheus also has good tool support by using the Prometheus Design Tool (PDT) and is viewed as one of the most mature AOSE methodologies Steghöfer et al. (2014).

Evaluating the methodology against the criteria set:

1. **Organisational structure for MAS.** Partly, as it does use the organisational concept of role to describe part of the agents' behaviour, and during the analysis phase roles are grouped to define agents. However, the roles are not part of the system. Uez and Hübner (2014)
2. **Heterogeneity.** No, as the internal design of the agents is defined as part of the methodology.
3. **Competitive and deceitful agents.** No, as the model is agent focused, and does not deal with deceitful agents.
4. **Open system** No, as the agent members are defined during the architectural phase and there is no mechanism for accepting new agents, or for dealing with agents leaving the system.
5. **Self-organisation** No, to both cases below as the top down decomposition of the system is centralised.
  - (a) **Reassignment of roles.**
  - (b) **Evolution of roles.**

## INGENIAS

INGENIAS is an agent oriented software engineering methodology for Multi-Agent Systems development Pavón and Gómez-Sanz (2003) which provides three elements to support engineers:

1. A visual language for defining MASs.
2. Integration with the software development lifecycle, defining deliverables close to the Unified Software Development Process (USDP) and activities that produce them.
3. Development tools.

INGENIAS suggests five views, models, of the system:

1. Agent model which describes the agents members of the system, their roles, tasks, goals, and their initial mental states.
2. Interaction model. This view describes how the interaction between agents occurs; between which actors, following a specific protocol, and aiming for certain goals.
3. Tasks and goals model which describes the relationship between goals and tasks. It also defines what the inputs and outputs of the tasks are, and how the outputs affect the environment and the mental states of agents.
4. Organisation model that defines groups of system components, which tasks are executing in parallel, and what constraints exist for the interaction between agents.
5. Environment model which identifies the system's resources, managing entities for these resources, and how agents perceive them.

Evaluating the methodology against the criteria set:

1. **Organisational structure for MAS.** Yes, as part of the organisational model.
2. **Heterogeneity.** No. Part of the agent model requires the designer to specify the agents' roles, tasks, goals and initial mental states. This would make it difficult for agents designed separately to join the system, especially since the interaction model does not cater for interactions with entities outside the system.



3. **Competitive and deceitful agents.** Partly, as the methodology can partly model the notions of trust and effectivity.
4. **Open system** Yes. INGENIAS has mechanisms to allow agents to subscribe into an organisation, or leave it. It also has a feedback based monitoring mechanism and a mechanism for expelling non performing agents.
5. **Self-organisation**
  - (a) **Reassignment of roles.** Yes. While Isern et al. (2011) suggest that INGENIAS does not support organisational dynamics, Gómez-Sanz and Pavón (2005) provide a description of subscription and leaving mechanisms.
  - (b) **Evolution of roles.** No, INGENIAS does not have a role altering mechanism.

## PASSI

The Process for Agent Societies Specification and Implementation (PASSI) is a step by step methodology from requirements to implementation for building Multi Agent Societies Cossentino (2005). PASSI provides a five steps iterative process, which comprises of the following steps, or models:

1. System Requirements. This model provides the functional description of the system, defines the responsibilities of agents, creates roles based on these responsibilities, and specifies the capabilities of the agents.
2. Agent Society. The society model describes the roles, the knowledge and communication patterns of each agent.
3. Agent Implementation. This model describes the agents from an internal structure perspective as well as from a behavioural perspective.
4. Code. This produces the code required based on self-generated code created by the PASSI ToolKit.
5. Deployment. This model deals with the deployment of the Multi Agent System.

Evaluating the methodology based on the criteria set above:

1. **Organisational structure for MAS.** Yes, PASSI is designed to create MASs and produces UML-like diagrams for the organisational tasks.

2. **Heterogeneity.** No, as the internal structure of agents is part of the design process. Also, the PASSI toolkit is required to develop agents based on the specification.
3. **Competitive and deceitful agents.** No, since agents' internal architecture is part of the design process.
4. **Open system** No, as the PASSI model has no independence from Architectures or Tools Steghöfer et al. (2014) and it does not provide any mechanisms for joining or leaving the system.
5. **Self-organisation** No, this is recognised as a future requirement by the authors of PASSI Cossentino (2005).
  - (a) **Reassignment of roles.**
  - (b) **Evolution of roles.**

## PROSA

Product Resource Order-Staff Architecture (PROSA) Van Brussel et al. (1998) is based on the concept of autonomous co-operating agents, called 'holons' and it's aim is to provide systems with adaptive and flexible control mechanisms.

PROSA is based on the idea of cooperating holons, where a holon is defined as "An autonomous and co-operative building block of a manufacturing system for transforming, transporting, storing and or validating information and physical objects. The holon consists of an information processing part and often a physical processing part. A holon can be part of another holon" Van Brussel et al. (1998)

PROSA defines three different types of basic holons:

1. Resource holon. A resource holon contains a physical part, i.e. a resource, and an information processing part, which controls that resource.
2. Product holon which defines the process, and contains the knowledge around the system's requirements.
3. Order holon which represents a task in the process.

The authors also suggest the following steps for their model:

1. Aggregation. This is action of grouping holons into other holons; similar to the idea of encapsulation where holons are consisted of other holons. Holons that can create a sub-system are aggregated into a holon.

2. Specialisation which further separates holons based on their characteristics.
3. Staff holons, which are more complex than the basic holons described above, are introduced to act as decision support mechanisms for the basic holons.
1. **Organisational structure for MAS.** Yes. The model is focused on MASs for manufacturing control, and it views the actors of the process as a recursive holonic organisation.
2. **Heterogeneity.** Partly. According to the authors, the decoupling of agents and holons in the reference architecture should enable a reuse of sub-systems throughout the industry. However, since PROSA is designed only for manufacturing control this limits the re-usability to other systems.
3. **Competitive and deceitful agents.** No, the model does not provide mechanisms for dealing with competitive or deceitful holons.
4. **Open system** Partly. The system is partly open in the specific domain as the end architecture has a high-degree of self-similarity Van Brussel et al. (1998) which means that new components should be easy to join the system.
5. **Self-organisation** No, the methodology provides no ways of designing reassignment or adaptation of roles during runtime.
  - (a) **Reassignment of roles.**
  - (b) **Evolution of roles.**

## ADELFE

ADELFE is a methodology designed to develop software according to the Rational Unified Process (RUP) and based on the authors' Adaptive MAS (AMAS) theory Bernon et al. (2002). ADELFE is not a generic AOSE methodology, it is focused on delivering self-organising MASs based on the AMAS theory.

The AMAS theory is based on the idea that a system's global function emerges from the collective behaviour of its members. This means that a global organisational structure is not coded into the agents members of the system. Also, since MASs need to be able to adapt their actions in order to cope with changes in the environment, thus dealing with open environments, it is the agents' self-organising nature that drives the reorganisation of the system. The

agents are required to be able to identify Non Cooperative Situations (NCS) and remedy them via their actions.

The ADELFE methodology has the following workflows which are sequential but can be reassessed in an iterative way:

1. Requirements workflow which produces the environment model, and contains five steps: to define the studied system, determine the context, determine the entities, characterise the environment of the system, and express the use cases.
2. Analysis workflow which defines the agents based on the entities identified during the requirements gathering phase. Analysis contains four steps: Identify the components, verify if the AMAS theory is adequate for the problem, identify the agents, and study the interaction between the different components.
3. Design workflow which produces the final design following four steps: To express the detailed architecture and agent model, provide agent architectures, express the NCSs, and provide class diagrams.

Evaluating the methodology based on the criteria set:

1. **Organisational structure for MAS.** Partly, through the design of agents and Non Cooperative Situations
2. **Heterogeneity.** No, as ADELFE designs the internal architecture of agents during the process.
3. **Competitive and deceitful agents.** Partly. The AMAS theory provides ways of dealing with NCSs. However, the fact that the agents' internal architecture is predefined does not leave room for deceitful agents to join.
4. **Open system** Partly. AMAS is aimed at describing open self-organising systems. However, the fact that the design workflow produces the detailed architecture of agents limits the openness of the system.
5. **Self-organisation** Partly. ADELFE does not use the notion of role when describing a system. The mechanisms for dealing with NCSs means that there is an element of agents re-planning their actions. (Bernon et al., 2004) note that the ADELFE model does not use the notion of role, as assigning specific roles to agents and imposing an organisational structure does not allow the system to be adaptable.

(a) **Reassignment of roles**

(b) **Evolution of roles**

## **IDE - EII**

IDE-EII, the Integrated Development Environment for Electronic Institutions, is focused on developing open heterogeneous MASs. IDE-EII consists of five different tools which deal with different steps of the software development process Arcos et al. (2005):

1. ISLANDER is a graphical tool used for the design and verification of the electronic institution.
2. SIMDEI, a simulation tool which is used to simulate and analyse specifications produced using ISLANDER.
3. aBUIDLER which is an agent development tool that creates agent skeletons using ISLANDER specifications.
4. AMELI, a software platform for running Electronic Institutions specified with ISLANDER.
5. a Monitoring Tool which allows the monitoring of the Electronic Institutions while being ran by AMELI.

The theoretical background of the ISLANDER environment is based on the concept of Electronic Institutions (EIs), which are designed using ISLANDER Esteva et al. (2001). EIs are seen as containing the following core notions:

1. Agents and Roles. Roles are defined as standardised patterns of behaviour, while agents are actors of the roles.
2. Dialogic framework. The dialogic framework establishes the acceptable communicative behaviours. These are based on speech acts, similar to real-life institutions where a common vocabulary, language, and conventions build the acceptable dialogues.
3. Scene which are well defined group meetings of agents, with defined communication protocols.
4. Performative structure which is a network of interconnected scenes. The performative structure captures how scenes relate to each other and how agents can transition between scenes.

5. Normative rules which are the rules that define how the agents' consequences and possibilities are affected based on their behaviour within the performative structure.

Reviewing the model against the criteria set above:

1. **Organisational structure for MAS.** Yes. IDE-EII supports organisational-inspired programming and has moved away from machine-oriented MAS views Sierra et al. (2004).
2. **Heterogeneity.** Yes, as the Electronic Institutions designed are "normative environments where heterogeneous (human and software) agents can participate" Arcos et al. (2005).
3. **Competitive and deceitful agents.** Yes, through the normative rules set on the EII.
4. **Open system** Partly, as ISLANDER focuses on the macro-level (societal) aspects of agents, not in their micro-level (internal aspects). One restriction is the definition of scenes/dialogic framework as this presumes that every agent willing to join the organisation is aware of them.
5. **Self-organisation**
  - (a) **Reassignment of roles.** Yes, as agents can join different scenes playing different roles.
  - (b) **Evolution of roles.** No, members of the EII need to play the specified roles, and there is no mechanism for evolving or adapting a role.

## OperA

Organizations per Agents (OperA) Dignum et al. (2002a) describes a MAS as an organisational structure regulated by social contracts, which describe the roles in the society, and interaction contracts which describe the interactions between agents. Isern et al. (2011). OperA is based on the theory of Electronic Institutions discussed above Garcia et al. (2009).

OperA defines the following models as part of the framework:

1. Organisational model. This defines the social structure (roles, groups, and dependencies between them), the interaction structure (scenes, scripts, connections between the scenes, and transitions between scenes). the normative structure (norms pertaining in roles, scenes, and transitions), and the communication structure.

2. Social model which defines role negotiation scenes and social contracts, which describe specific agreements for role enacting agents.
3. Interaction model which describes the interaction protocols realising contracts and scenes.
1. **Organisational structure for MAS.** Yes, OperA is focused on designing MASs from an organisational perspective.
2. **Heterogeneity.** Yes, as OperA describes agent societies in terms of their structure without being concerned with the way individual agents are designed Dignum et al. (2002a).
3. **Competitive and deceitful agents.** Yes, as the designer does not control the agent design and the individual goals, allowing for competitive and deceitful agents that can alter the society's behaviour by not conforming to the intended behaviour Dastani et al. (2004).
4. **Open system** Partly, as OperA does not define or presume the agents' internal architecture, but the restriction of a known interaction model that was identified for ISLANDER exists for OperA as well.
5. **Self-organisation**
  - (a) **Reassignment of roles.** Yes, as actors can join different scenes and performing different roles.
  - (b) **Evolution of roles.** Partly, as the model allows for agents to bring individual behaviours and tasks to the roles Dastani et al. (2004). However, the model does not provide a mechanism of incorporating these new behaviours and making use of them by adapting or evolving roles.

## Moise

The Model of Organisation for multiI-agent SystEms (Moise) deals with the design of MASs by combining agent-centred approaches and organisation-centred approaches Hannoun et al. (2000).

The Moise model contains three levels:

1. The individual level which defines the responsibilities of each agent and the roles. A role is seen as a set of missions that an agent playing a role must obey. A mission is a set of constraints that an agent executing a task must take into account.

2. The agency level which aggregates agents into larger structures, groups. The groups are part of the organisational entity of the system and contain roles, missions, and links.
3. The society level, which provides the global structure and the interconnection, links, between agents and structures. Links structure the social exchange between roles, and can be of a communication, structuring, or authority type.

Moise defines the organisational structure as a set of roles, groups, and links between them.

The Moise+ model (Model of Organization for multi-agent SystEms) (Hübner et al., 2002) also provides three levels of abstraction, the structural, functional and the deontic aspects. It uses the notions of groups and roles, as well as the notion of links between roles describing the communication patterns. The deontic aspect defines the obligations and permissions of each role-holder with regards to accomplishing their missions within a role.

1. **Organisational structure for MAS.** Yes, as the model combines an organisation-centred and an agent-centred approach
2. **Heterogeneity.** Yes. Moise does not prescribe the design of individual agents.
3. **Competitive and deceitful agents.** Yes, the model caters for competitive and deceitful agents through its society level.
4. **Open system** Partly, as the model does not exclude new members from joining the organisation, but it does not provide mechanisms to do so.
5. **Self-organisation** No, as Moise does not cater for reassigning roles, or adapting roles during runtime.
  - (a) **Reassignment of roles.**
  - (b) **Evolution of roles.**

### Communicating open systems

D’Inverno et al. (D’Inverno et al., 2012) provide a very detailed definition of electronic institutions and, through this, the basis of a framework for designing open MASs.

After providing background information on institutions, the authors give an informal description of electronic institutions. Then, they discuss the communication patterns needed for the agents to be able to cooperate in an open



system. The authors also introduce the notion of role which allows them to de-couple electronic institutions from agents so that the design of the institution does not depend on the agents' internal structure. Roles have the following characteristics:

1. They are de-coupled from the agents' internal structure. They are not influenced by the capabilities of each agent holder in the same manner that roles in a theatrical play are not altered by the actors that play the role.
2. Roles appear in predefined scenes and transitions.
3. They specify the role holders' behaviour and communication patterns.

After introducing the notions of 'scene' and 'scene transition', they provide formal data structures to describe scenes, scene transitions and utterances (points of communication between two agents in a scene). These data structures, in conjunction with the communication language, build up the framework of the electronic institution. Finally, the concept of the system's infrastructure, which plays a pivotal role in coordinating the agents joining and leaving scene instances or the system, is introduced. The authors continue providing a review of available specification languages and tools for providing a computational framework of electronic institutions. Using the tools identified, they pass from the descriptive framework to a computational framework and its design. Finally, they evaluate their method using existing approaches and they close the article discussing relevant work and open questions.

1. **Organisational structure for MAS.** Yes, the model is focused on designing electronic institutions, using the notions of roles that are played by agents, and scenes where agents participate.
2. **Heterogeneity.** Yes, since the model does not predefine the agents' internal structure.
3. **Competitive and deceitful agents.** Partly. The Electronic Institutions created are monitored by the infrastructure which ensures that actors behave correctly. However, it is not defined how the infrastructure would do so.
4. **Open system.** Partly. The model allows for agents to join and leave the organisation, but only as long as they are aware of the communication patterns and scenes.

## 5. Self-organisation

- (a) **Reassignment of roles.** Yes, as new agents can join the organisation, assume a role, and start participating into scenes.
- (b) **Evolution of roles.** No, the model does not allow for adaptable roles during runtime.

### 1.5.3 Summary

Reviewing the models and their evaluations, one can see that all models have their strengths and weaknesses. Also, most models can be used to design open MASs.

The table below summarises the results of the evaluation process.

Table 1.1: Evaluation of models

Model	Organisation and MAS	Heterogeneity	Competitive Deceitful agents	Open System	Role Reassignment	Role Evolution
JADE	Yes	Partly	Partly	Partly	No	No
Tropos	Yes	No	No	No	Partly	No
O-MaSE	Yes	No	No	Yes	Yes	No
Gaia	Yes	Yes	Yes	No	No	No
Prometheus	Partly	No	No	No	No	No
INGENIAS	Yes	No	Partly	Yes	Yes	No
PASSI	Yes	No	No	No	No	No
PROSA	Yes	Partly	No	Partly	No	No
ADELFE	Partly	No	Partly	Partly	Partly	Partly
IDE - EII	Yes	Yes	Yes	Partly	Yes	No
OperA	Yes	Yes	Yes	Partly	Yes	Partly
Moise	Yes	Yes	Yes	Partly	No	No
Communicating Open Systems	Yes	Yes	Partly	Partly	Yes	No

Reviewing the results of the evaluation, one can observe the following:

- All models are focused on supporting MASs with an organisational structure, or have evolved to do so, like Gaia. This confirms the literature that

organisation-centred MAS are seen as the obvious solution for bringing together autonomous agents into one system.

- Most of the models that support Open systems, fully or in part, also provide support for dealing with competitive and potentially deceitful members. The same stands for support of heterogeneous agents. This is in agreement with the analysis of Virtual Organisations from Luck et al. (2004) who suggest that the main issues when designing Virtual Organisations are: heterogeneity, dealing with trust and accountability, dealing with failures and errors, and societal change.
- Not many models support re-organisation during runtime. It is mainly the models based on the Electronic Institution theory that offer mechanisms for role re-assignment during runtime.
- No models offer full support for role evolution and adaptation during runtime. I believe that having predefined roles, which cannot be altered, in predefined groups, removes flexibility from the system. Different agents can join and leave the system bringing new characteristics and behaviour to the pool of available behaviours. Not catering for role evolution does not allow for new, potentially interesting behaviours to emerge. The concept of role creation is important in this thesis as it allows for completely new roles to emerge during runtime allowing for the system to adapt to changes in the environment and to changes in the members' behaviour.
- While most of the models offer support for agents to join and leave the organisation, focusing on open systems, I believe that they partly support openness due to the prerequisite for new agents to know the predefined communication patterns required. Speech acts are the obvious solution to design communication in an organisation, however they pose this limitation rendering the final system not truly open. This thesis will explore the notions of stigmergy and environment mediated communications as potential solutions to this issue.

## 1.6 Open questions

In the context of Agent Oriented Software Engineering, the focus of this thesis is how to describe the blocks of an organisational structure to be used on open, competitive, heterogeneous, self-organising MASs, and provide a descriptive framework for specifying these blocks when designing a MAS of this type;

using the secondary equity markets as an example. From a software lifecycle viewpoint, that contains four distinct phases of planning, analysis, design, and implementation, the thesis focuses on analysis and design.

This thesis will discuss the idea of providing open self-organising systems with an organisational structure, with this structure being defined in behavioural terms. In particular, it will discuss different methods of self organisation and will argue that a bottom-up decentralised approach is best suited for maintaining social order in open systems.

It will provide a formal definition of the notion of role in self organising systems, based on the argument that the role is the building block of an organisation (Parunak and Odell, 2001). It will be argued that a behaviourist approach is well placed to be used in defining the notion of role in a self-organising structure.

In addition, communications will be investigated as one of the basic characteristics of organisations, with focus on environment mediated communications and behaviour based communications. Having analysed these two notions, it will investigate how social order can be achieved and defined in behavioural terms.

Despite the fact that several working MASs have been designed and developed, several questions remain open with regards to specifying open, self-organising MASs. The following issues are of central interest for this thesis:

1. The characteristics an organisation needs to have in order to self-adapt its structure. This thesis will examine methods of self-organisation and will argue that a bottom-up, decentralised approach needs to be followed in order to design open MASs.
2. How the notion of role can be formally defined in order to design open, heterogeneous, self-organising MASs. The thesis will compare the so-called behaviouristic and mentalistic approaches in specifying roles in an organisational context and will aim to define the notion of role in strictly behavioural terms. This thesis will argue that a behaviouristic approach can provide the flexibility needed to design organisational structures with the ability to reassign and evolve their roles.
3. How can environment mediated communications be incorporated into the design of open, heterogeneous, self-organising MASs. The main focus of the thesis on this point is that communication based on speech acts limits the openness of a system, and that environment mediated communications can offer an interesting approach to designing communication patterns in an organisational structure.

4. Based on the definition of role in behavioural terms and using environment mediated communications, this thesis will produce a descriptive framework for designing open, self-organising, heterogeneous, competitive MASs in behavioural terms, using the secondary equity markets as an example.

## 1.7 Summary of contributions

This thesis provides the following contributions:

1. It argues that the building blocks of an organisational structure can be defined based on behavioural terms. It provides a list of characteristics that the definition of role (chapter 3), communication (chapter 4) and social order (chapter 5) should meet and also provides a definition of these notions in behavioural terms (in the relevant chapters).
2. These definitions are combined into a descriptive framework which can be the basis of a formal specifications framework for designing open self-organising systems in behavioural terms (chapter 6).
3. This framework is, then, applied to the example domain of secondary equity trading markets, providing a high level description of that domain in behavioural terms.
4. An expansion is given to the definition of informational content as provided by Skyrms. This definition can be used to define when a signal contains information about the environment (section 4.6).

## Chapter 2

# Organisational Structures and Self Organisation

This chapter discusses the view of imposing an organisational structure on open systems, how this structure can be designed to be adaptable and how it can be represented based on a behaviouristic approach.

### 2.1 MASs as organisations

It is suggested that social ability is one of the main characteristics of agency (Odell, 2007a). In a MAS, the agents do not act as isolated entities. They need to work in groups capable of performing complex tasks in order to achieve their goals. To form these groups, agents need to communicate, to co-ordinate and control their work, to delegate and prioritise tasks. In other words, the agents that participate in a MAS need to have 'social behaviour' either because they have to accomplish a common goal or because their interaction with other agents helps them accomplish their own goals (Zambonelli et al., 2001). The fact that the above characteristics can be found in human organisations, reiterates the need to design MASs using paradigms and structures from sociology and organisational theory. According to Gasser (1992), an organisation can be defined as follows: "An organisation provides a framework for activity and interaction through the definition of roles, behavioural expectations, and authority relationships (e.g. control)."

In addition, when designing a MAS, one needs to take into account two divergent characteristics of the system, the agents' autonomy and the system's global purpose. MASs are, usually, designed to have a global purpose, a function

they need to accomplish or a service they need to offer. For example, a large MAS for manufacturing control would aim to produce more products, at a specified standard and in less time. On the other hand, agents are autonomous entities with proactive behaviour (Zambonelli et al., 2001). Agents are able to reason, decide and act outside a common flow of control taking initiatives when necessary to achieve their goals. Belonging to an organisation poses limits to the agents' autonomy, as they have to comply to the organisation's constraints (Sichman et al., 2005). This reasoning also leads to the idea that MASs need to have an organisational structure.

An organisational structure would also assist in predicting the system's behaviour. In a MAS, and in any society-like system, complex dynamics appear. The members of the system, being autonomous and intelligent in nature, tend to form relationships and act in an unpredictable way (Jonker and Treur, 2009). Due to these complex dynamics, it is difficult to predict the system's behaviour. As a result, an administrator or observer of the system cannot have a clear picture of the system's state and, more importantly, cannot predict its behaviour. In addition, the agents themselves, not being able to predict the system's properties, cannot make informed decisions. Researchers suggest that these complex dynamics can be managed by imposing an organisational structure on the system (Vazquez-Salceda et al., 2005) as they reduce complexity, increase the system's efficiency and make it more scalable Abbas et al. (2015).

Moreover, an organisational structure provides the ability to take advantage of emergent behaviours. The autonomy of the agents and the complex social interactions can lead to desirable, but difficult to design, behaviours. Especially in open systems, it is almost impossible for someone to predict all possible interactions and permutations of actions. As a result, there might be cases where unpredicted behaviours might emerge. An organisational structure allows for the identification of helpful properties and allows for incorporating them into the structure (Ferber et al., 2003).

In addition, as Multi Agent design is often used in medium or large scale systems, the design needs to allow for the system to be scalable. In an open MAS, agents are allowed to join and leave the system and this should not be affecting the system's overall performance. The system needs a way of controlling what happens when a new agent joins the system, or when an agent leaves the system. Also, as in a real world example such as the secondary market, the agents are usually designed and developed by different organisations (Ferber et al., 2003). A high level organisational structure is required to ensure that the rules of the system are followed and that the aims and the strategies of the MAS are not

altered, or are altered in a controlled way, by the agents' availability.

Finally, seeing the system as an organisation, with defined roles and relationships, allows for the creation of groups with distinct boundaries between them. An organisation usually has different groups performing different tasks, with these groups being treated as black-boxes. Groups can interact with each other, and members of the same group can know the internal mechanisms of it, but don't need to know the internals of other groups. This level of abstraction allows for:

- Flexibility; since not all agents have to be aware of the mechanics of all groups, agents can leave and join the system more freely.
- Security; since a group of agents can be seen as a black-box, it is more difficult for agents outside the group to take advantage of this group (Isern et al., 2011).
- Decentralised design; since some agents are hidden from others, one does not need to assume a standardised architecture or implementation (Isern et al., 2011).

Researchers and designers of applied MASs have used several organisational structures to specify and control the behaviour of the system. These models vary from strict hierarchical structures that use a command and control approach (Van Nieuwenborgh et al., 2007) to network models, holarchies, coalitions and societies (Hörling and Lesser, 2005). These approaches can be categorised into mechanical and organic (Isern et al., 2011). When referring to mechanical approaches, one means organisations where every interaction and behaviour are bound to strict rules and protocols. An organic approach is one where interactions are specified in a more collaborative approach, at a local level, without strict control.

To design a multi agent system, one needs to define how the agents form this system. As agents are autonomous, rational and social entities, it seems reasonable to use human organisations as an analogue and a design paradigm for multi agent organisations. This approach has been followed by most of the prominent AOSE methodologies over the past years. An organisation can be described by four attributes: its high-level function, the roles that exist in it, the methods of communication and the method to assess if it accomplishes its purpose, its evaluation function (see for example, (Sims et al., 2003)).



## 2.2 Holism

As discussed above, one of the reasons for using organisational structures is the need to manage the system's behaviour and to combine the system's high level purpose with the agents' autonomy.

Based on the sociological theory of holism, some researchers suggest that the behaviour of a MAS emerges from the behaviour of the participating agents through complex dynamics. These dynamics are created through 'social' interactions between the communicating agents in ways that are different than a simple collection of the agents' individual behaviours (Schillo et al., 2002). This is not to say that the system has properties that can not be explained and induced from the components' behaviour and interactions. It means that the autonomous behaviour and the interactions between these components are so complex that it is very difficult to predict the system's behaviour analysing each and every agent and its interactions with other agents.

Holism is defined by Smuts as "the tendency in nature to form wholes that are greater than the sum of the parts through creative evolution" (Smuts, 1952). The concept of holism has induced some schools of thought in several sciences to examine a given system not only by analysing the system's parts and their relationships, but also by perceiving the system as a whole. MASs can be analysed based on their components' behaviour, while also accepting that this behaviour can be so complex and unpredictable that the system's behaviour as a whole needs to also be analysed separately. Analysing the system from a top-down view, and breaking it down into sub-systems, 'holons', which in turn contain other holons has also been suggested by Van Brussel et al. (1998) for their PROSA framework. Holons are at the time self-contained wholes to their subordinate parts, and dependent parts to their higher level holon.

This approach, designing the system as a whole based on the agents' behaviours, can offer some interesting characteristics to the system produced, such as encapsulation and granularity. Encapsulation is a design principle, used among others in Service Oriented Architecture, which states that the components' implementation should not be exposed to the environment (IBM, 2007). Designing a MAS as a whole, follows this principle and it can benefit the final design, decoupling an agent's implementation and its behaviour at a high level view allowing for re-usability and easier system maintenance. The characteristic of granularity is widely used in system design and it means that the components' behaviour can be combined to a higher level group behaviour depending on the level of detail one needs to examine. A holistic approach will follow this design principle viewing the system as a whole that can be broken down into smaller

parts, groups of agents, that can subsequently be decomposed in smaller groups of agents or the agents themselves. A similar design is suggested by Schillo in (Schillo et al., 2002). Schillo views the agents as ‘holons’, entities that are viewed as simple agents by the other components but can in fact be groups of agents themselves.

## 2.3 Self-organisation

Designing large-scale systems as MASs based on organisational structures helps to control the agents’ autonomy, to predict and manage the system’s behaviour and also makes the system scalable. However, as in human organisations, the system needs to be flexible enough to adapt its structure to changes in its environment, to changes in the members’ behaviour, and to members joining and leaving the system. An organisation is not a stand-alone entity, it interacts with other organisations and with its environment. This interaction often triggers changes to the organisation itself; some times imposing a change on its structure. A software development company, for example, might identify that their products are unstable and do not offer the functionality expected. This identification could come from the company’s clients, complaining for the performance of an application, or from inside the organisation, based on its own quality assurance procedures. This observation could lead to a change in the organisation’s structure; it can be decided that more testers need to be employed, that the development procedure needs to be altered or that a new office for handling complaints needs to be set up. As in the example of the human organisation above, there will be cases when an open MAS will need to identify that a structural change is needed and will need to perform this change.

### 2.3.1 Reasons for re-organisation

The ability to adapt is crucial for MASs as many factors affecting the system can change during runtime (Di Marzo Serugendo et al., 2006) and (Vazquez-Salceda et al., 2005).

The environment in which a MAS operates is dynamic and difficult to predict. As MASs operate using computer networks, and in the general case interacting with humans and each other, their environment is subject to frequent changes. Network connections can close and new connections can be established, agents might become unavailable and communication and work patterns might differ in different networks. The environment is not a passive entity, it interacts with and affects the system. Changes in the environment might oc-

cur that cannot be dealt with by the existing organisational structure. In this case, this structure would need to be redefined so that the system can continue functioning.

In addition, as in open systems agents can join and leave during runtime, it is expected that the structure will need to be changed during runtime. For example, some of the participating agents that form a certain part of the organisation might leave the system at once. This would cause specific roles, and the behaviours associated with them, to cease to exist. In a holonic view of the system, one would only need to identify agents that can perform the functions that the previous members of the holon were performing. Another option would be for a new structure that offers the same results to be adopted.

Moreover, new agents with different characteristics might join the system. This could trigger a change in its structure; for example a new agent with more capabilities could perform a task currently performed by a group of agents and free them to create new groups performing different tasks. On the other hand, an agent's performance might deteriorate over time. This would call for another agent to assume its role, or for a new local structure to be created to perform the same functions.

A recent example in the secondary equities market sector is the introduction of crossing networks as available venues of execution. The past few years there has been a decrease to the volume of shares traded due to the global financial conditions (change in the environment). This had created difficulties in trading large blocks of shares. Brokers are sceptical about trading large blocks in the regulated markets, as this will have an adverse effect on the price of the stock. For this reason, a few large brokers have set up crossing networks (new roles), offering market-like facilities in an anonymous way and without disclosing pricing information prior to the trades being confirmed.

### **2.3.2 Reasons for self-organisation**

A predefined, inflexible structure would not allow the system to adapt to changes discussed above. Additionally, altering the structure of the system exogenously would require a large amount of human resources; an approach neither practical nor cost effective. For an administrator of the system to be able to restructure the organisation, they would need to be able to:

1. Identify that re-organisation is needed.
2. Have a good overview of the structure at the given time.
3. Plan a change based on the reason for re-organisation and the structure.

#### 4. Implement the change to the system

This task becomes more difficult when talking about open systems, where the internals of agents are not necessarily known. Moreover, this task needs to occur quickly to ensure that the system is not affected and keeps operating as normal.

The above suggest that the system needs to be able to restructure itself without the intervention of a designer or of an administrator. Moreover, the agents' characteristics might not be known from the system design phase, as different designers might be involved in building the system (Di Marzo Serugendo et al., 2006). In a generic case, one system's agents would need to co-operate with other systems' agents which might be specified and developed in a different way. In addition, as the scale of the system becomes larger and its structure more complex, it is not possible to predict every interaction, every relationship and how these would benefit the organisation (Kephart and Chess, 2003).

The above illustrate that a MAS needs to have the ability to adapt to changes and alter its structure accordingly to continue performing its function. An open system needs to be designed having an organisational structure, but this structure needs to be flexible enough to allow for re-organisation. And this re-organisation needs to happen by the system itself, without the need for external intervention. The system needs to be adaptive, dynamically reacting to changes altering its structure. For example, in the case of a MAS designed to work in an hierarchical structure, if the agent representing the head of the structure fails, then the whole organisation would fail. In this case, a system that could adapt its structure by appointing a new head or by rearranging the hierarchy, would continue working without the need for external intervention.

The need for self-adaptation of the organisational structure means, in other words, the need for self-organisation. Ashby (1947) defines self organisation as the process whereby a system changes itself "from a bad organisation to a good one" (p.267). In the case of MASs, we would define a bad organisation as one that ceases to provide the function it was designed to as a result of the changes in its environment or its members, and as a good organisation one that continues to be functional despite these changes.

### 2.3.3 Approaches to self-organisation

The common characteristic of many of AOSE methodologies available is that despite the fact that they are used to specify complex, large systems of autonomous agents, the systems they produce do not allow for a great degree of flexibility or do not allow for self-organisation.

One could identify different ways of designing a dynamically self-organising system; from assigning the role of adapting the system’s structure to a single agent, to allowing agents to form groups according to their needs and let the overall structure emerge.

### **Top-down approach**

The first approach is called a top-down (Odell, 2007b), or centralised, approach where the designer of the system has delegated this task to some agents by giving them the power and the ability to restructure the system when needed. An example of this case would be a hierarchical MAS that uses a strict tree organisational structure with an agent delegated to review and decide the organisation’s structure. In this case, the agent will be receiving feedback regarding the system’s performance from the other agents and will have both the capabilities and the authority to restructure the system when it decides that this is needed.

A proposed framework that uses a top-down approach for reorganisation is MOISE+ (Hübner et al., 2002); (Hübner et al., 2006). In MOISE+, the agents send requests to a designated managing agent asking for reorganisation, for example, for the creation of new groups. The organisation’s manager is the agent who decides to accept or reject these requests. This model provides a centralised method of reorganisation. Its main advantage is that the organisation’s manager has absolute control on the changes made to the system and, therefore, the system’s administrator can know its structure at any given time and have greater control on malicious requests. Despite this advantage, this model requires the existence of an agent that has an overview of the whole system at any given time and is always capable of taking decisions about its reorganisation. This task would become increasingly difficult as the system grows. In addition, the flexibility required is absent as the dynamic reorganisation of the system is based on the bottleneck created by a sole managing agent with more capabilities than the other agents, meaning that if the managing agent becomes unavailable the mechanism would fail.

Top-down approaches where designated agents have decision making permissions and capabilities over the system’s structure, for example in an hierarchical approach, are fragile. These agents constitute single points of failure and make the system vulnerable to accidents. Moreover, under normal operating conditions, they can cause bottlenecks slowing down the system’s performance (Isern et al., 2011) and (Parunak, 1997). Even if the designated agents are designed for coping with a larger amount of requests, they still pose a limit to the system’s performance; they are a boundary that cannot be exceeded (Parunak, 1997).

### **Bottom-up approach**

The second approach for self-organisation is what one would call a bottom-up approach (Odell, 2007b). The agents' behaviour and the way they relate are governed by certain rules but it is the agents themselves who communicate and determine the new structure of their organisation at a local level, not a managing agent. It is the local interactions that influence the structure of the system at a local level and, as a result, its global behaviour (De Wolf and Holvoet, 2004). This approach mirrors the decentralisation that exists in the internet and is becoming common in today's business world. It assumes that there is not one authority that specifies and controls every change, for example, the administrator of the system or the head of a company; on the contrary, it is the agents that decide the best way to react to a change and reform the organisation based on specific rules.

In this approach, neither the administrator nor a delegated agent control the way the organisation reacts to changes. The agents interact, co-operate at a local level to locally reorganise themselves and, as a result, create a new organisational structure that is capable of working under the given circumstances (Bedau, 2003). The communication channels can change, different roles can be assigned to different agents and, whenever the need for a structural change emerges, new roles and groups can be created.

An example of using a bottom-up, decentralised, approach to reorganise structures can be found in artificial economies, which are agent-based models of economic systems, where the artificial macro economic regularities evolve from the local interaction of the economic agents governed by internal procedures (Bedau, 2003). As a result, market wide regularities, such as price structures, change not due to a centralised reform but due to the agents' local interactions. A bottom-up decentralised model of self-organisation is also used in the T-Man network protocol. The nodes of the network decide their neighbours using metric algorithms and then, through information propagation and ranking, the whole network structure is reorganised (Di Marzo Serugendo et al., 2006). Simulations of this protocol have shown that after a change occurs, for example when a large number of nodes becomes unavailable, the network is restructured and eventually becomes stable (Jelasity and Babaoglu, 2006).

Based on the idea of holism described in section 2.2, and since MASs are a form of collective intelligence in the sense that their characteristics emerge from the components' characteristics (O'Reilly and Ehlers, 2006), one needs to take advantage not only of the individual agents' characteristics but also of the system's emergent characteristics. The system should be allowed to evolve and have

a mechanism of evaluating these emerging attributes. Designing MASs to work in an organisational context provides the tools for controlling and observing the system's structure and behaviour. However, I believe that this organisational structure needs to be maintained by the agents themselves for two reasons. Firstly, this bottom-up reorganisation has the potential to reveal new, difficult to predict and design, emergent characteristics, and secondly to achieve complexity in conjunction with the flexibility required. The bottom-up approach has the disadvantage of low predictability, that is, the emergent organisational structure cannot be easily foreseen. It offers, however, a real decentralised way of dynamically reorganising the system's structure.

## 2.4 Behaviourism

Except for the actual organisational structure and how this can self-adapt to changes, it is important to define how this structure is represented. One needs to describe the organisational structure and its building blocks using a method with the following characteristics:

1. The method needs to have the ability to describe the required notions and ideas.

When describing an organisation, one needs to describe how the members work together, how they communicate and how social order is maintained. These notions are complex and their definition can contain other complicated and ambiguous notions. The way of describing the organisation needs to capture notions such as decision, power, trust, deceit, importance, etc...

2. The method should be expressed in such a way so it can be translated into a specification.

The organisational structure produced by a framework will need to be applied to software agents. As a result, this structure needs to have a logical way of representing notions. The description of the structure needs to be done in such a way that provides an unambiguous method of designing and building open self-organising systems.

It needs to be reiterated that this thesis is not trying to produce a detailed methodology for designing open self-organising MASs. It produces a descriptive definition of the building blocks of an organisational structure that can be applied on such systems. However, it is important for this framework to be based on the same principles. This section introduces behaviourism, the method of

psychology, and discusses how this will be used to represent the building blocks of an organisation in the framework suggested.

Behaviourism as a doctrine of psychology has its roots in the early 1910s, when John Broadus Watson declared that psychology's theoretical goal is the "prediction and control of behaviour" and that introspection is not a scientific method for understanding behaviours. The main goal of behaviourism is the prediction and control of observable behaviour, as this changes based on environmental stimuli.

Ivan Pavlov's experiment is a classic example of the behaviourist mindset. In his experiment, Pavlov had trained a dog while feeding it. Pavlov noticed that, when food was offered to the dog, it was salivating (conditional stimulus). Pavlov was also ringing a bell before presenting the food to the dog. He noticed that, even when he was ringing the bell without offering food, the dog was still salivating (unconditional stimulus). Pavlov termed this result 'classical conditioning through association'.

Edward Thorndike expanded Pavlov's experiment, discussing about instrumental conditioning and learning. In his experiment, he placed a cat in a cage with food outside of it. Thorndike observed that, after the cat managed to get out of the cage by opening the door, then the frequency of this occurring increased. With this experiment Thorndike showed that behaviours have positive consequences through environmental stimuli. The fact that the cat opened the door, initially by a random action, led to the cat being fed. As a result, the behaviour of opening the door became more frequent, showing that the cat learned how to acquire food in that particular scenario.

B. F. Skinner also studied behavioural reinforcement. In his experiment, Skinner held rats in a cage which had some levers. Some of the levers were providing food to the rats when pressed. Skinner observed that after the rats pressed the correct levers and received food, the frequency of pressing them increased. Moreover, when the function of the levers changed and stopped delivering food, Skinner observed a decrease to the lever pressing frequency. It is Skinner who, from a philosophical viewpoint, suggested that mental terms exist as an effort to explain behaviours. For example, when one says that they believe in A, they actually mean that in a situation relating to A, they would behave in a specific way. According to Skinner, mental states can be translated into behavioural terms.

This thesis is not trying to examine behaviourism as a method of psychology. The debate between behaviourist and cognitive psychologists is interesting for this thesis, has been long settled, and is out of scope of the final framework.



This thesis examines the use of notions extracted from behaviourist psychology in the fields of information theory, of artificial intelligence and of computing. Behaviourism provides interesting ideas that can tackle a few issues that exist in these sciences. It poses, nevertheless, limitations that do not exist if one follows an approach inspired by cognitive theories. However, as it is argued below, these limitations can be overcome. The gains from using a behaviouristic approach are very interesting and need to be explored.

### **2.4.1 Notions**

When one describes organisational structures and methods of interaction, it is difficult to avoid using notions that have a mental background, that refer to states of mind. Several notions used throughout this thesis have, instinctively, a mental background. This is expected and, some times, required. Language contains many notions which are not strictly defined in behavioural terms and are used in every day life. Every day, one makes decisions, feels that the weather is cold, believes that they are running late for work and thousands more examples. Some of these notions are used throughout this thesis as well. As it provides a descriptive framework, the use of everyday language is inevitable and, hence, the use of such type of notions hard to avoid.

In addition, this thesis follows the behaviourist paradigm in the sense that all the notions needed to describe a self-organising system can be expressed in behavioural terms. This does not suggest that mental notions will not be used in the description of the framework, it is to say that these notions will be defined and explained in behavioural terms where appropriate. The following sections discuss the notions most often referred to in the thesis in behavioural terms.

#### **Decide/decision**

The notion of decision is prevalent in real life examples as well as in the design of intelligent systems. Members of an organisation confront situations where they have to decide their next action. For example, when a change in the environment occurs, when something is communicated to them, or simply when it is the time to make a decision. It is accepted that agents, being rational entities, will have a decision making process. This process can be from a simple control process, if A happened, then perform B, to a more complex process based on learning and prediction algorithms.

In a behavioural view of an organisation, this decision making process is irrelevant to the performance of the organisation. Agents can decide to act,

or not act, in specific circumstances, but what is important is to observe their behaviour under these circumstances. Each agent can behave in the expected way or not, based on their decision. The organisation needs to have mechanisms to control unexpected and unwanted behaviours, but this can be done based on what is observed, not based on how this was decided. The way an agent decided to act, its internal decision process, is irrelevant to the organisation.

### **Understand**

This notion is used when discussing the communicational aspect of organisations. Agents transmit messages and an important aspect of the communication process is if the receiving agents understand the message. Furthermore, an agent can understand sensory input. If the environment changes in any way, agents can realise that something has changed. This is based on their ability to understand sensory input.

In a behaviourist approach, the notion of 'understanding' can be bypassed. If an agent wants to make another agent aware of an event, this can be observed on the receiving agent's behaviour. It is irrelevant to try to define if the receiving agent understood the message as long as this agent behaves in the expected way. This behaviour could be the result of the agent understanding the message and acting accordingly (most probably), or of the agent randomly deciding to act in that specific way. In either case, the message had the same effect, the expected behaviour. If, in the same example, the receiving agent does not act as expected, it could be the case that they didn't understand the message or that they did but decided to not act. Again, in either case, the important event is that the agent did not act as expected. The reason why this happened is irrelevant to an observer or other members of the system.

### **Importance**

This notion is used primarily in chapter 4 to describe important states of the environment. One can argue that the notion of 'importance' is a mental notion. One deems a piece of information important if it helps them in their goals, or if they think that it is important and this is seen as a subjective decision.

This thesis argues in section 4.6 that the importance of a signalling message can be defined based on information theory.

## Deceit

The notion of deceit is important when one describes organisational structures. Deceit can potentially exist in every form of communication. There are multiple examples of deceit in human organisations, one can oversell their product or an employee might fake illness.

In the equities market example, a broker might try to deceive their counterparts by placing fake orders in the market. It is a practice, illegal nevertheless, of some brokers to try and influence the price of a stock by entering and quickly withdrawing orders of the opposite direction. For example, if a broker wants to sell a stock in price A, they might place buy orders at a price close to A which is higher than the existing buy price, and withdraw them quickly. Other brokers who are monitoring the price of a stock will believe that the price is going up, and they will start following this trend. When the price of the stock reaches A, the deceiving broker will place their real order which will execute.

Searcy and Nowicki (2005) define deception as follows (p.5):

1. A receiver registers something Y from a signaller.
2. The receiver responds in a way that benefits the signaller and is appropriate if Y means X but
3. it is not true that X is the case.

Most researchers define deception based on other mental notions. Deceit exists when the sender of a message alters the receiver's beliefs in such a way that these beliefs are wrong but benefit the sender. Searcy's definition however does not follow a mentalistic approach. It is irrelevant what the beliefs of the receiver are or what the intentions of the sender were. Searcy's definition of deceit follows a behaviourist approach.

## Expectation

The term expect (expected, expectation) is used in this thesis and in MAS models when referring to behaviours or actions that should happen, based on specific circumstances and understanding of the system's status. This could be the behaviour that an observer or the designer expects, or that other agents expect. When referring to the behaviour that an observer expects, we mean the expected behaviour according to the plan, according to the design of the system.

It is harder to define what expected behaviour means when referring to the behaviour that agents expect other agents, members of their local environment,

to have. Using this term, one could talk about beliefs (what the agent believes that the other agent should do) and intentions (what the acting agent intended to do). In a behaviourist approach, and since behaviours are observed based on the environmental change they produce, the expected behaviour would be described as the behaviour that produces the most probable outcome.

In the market manipulation example above, the expected behaviour on behalf of the other participants would be for the price of the stock to raise. This is the behaviour expected based on market norms, buy low - sell high. This is also what is expected based on previous experience and based on what would seem sensible. From a behaviourist organisational point of view, it is irrelevant why the deceiving broker expects this behaviour (possibly because of all three reasons).

## 2.5 Organisational model

The sections above discussed the need to impose an organisational structure in order to control the agents' autonomous nature, provide guidance to an open system and improve visibility to an external observer. The notion of holism was discussed showing that viewing different groups of agents as black-boxes provides the system with interesting characteristics which are required by current software development techniques: reusability, and granularity. Furthermore, it was shown why the organisational structure of an open multi agent system needs to be able to self-adapt without external intervention. Methods of self-organisation were discussed and a bottom-up approach was found to be preferable. Finally, the notion of behaviourism was introduced. It was discussed why behaviourism can offer an interesting alternative to mentalistic approaches and several widely used notions were discussed in behavioural terms.

Based on the above, this thesis provides a descriptive framework for designing organisational models with the following characteristics:

1. The organisation is defined based on four building blocks: the high level purpose, the roles that exist in it (chapter 3), the communication method (chapter 4) and a mechanism for maintaining social order (chapter 5).
2. This structure can self-adapt, it can change without external intervention. This makes it ideal for being applied to open systems of autonomous agents.
3. The building blocks are defined following a behaviourist approach, without referring to mental notions or the agents' internal mechanisms.

## Chapter 3

# Roles

### 3.1 Organisation and roles

Role playing ensures co-operation, or at least co-existence in the same environment, between the different members of a society (Burton, 1969). In human organisations, different roles are needed to collectively achieve a goal. Roles are designed and assigned to members of the society in almost every form and level, in the family, at school, in a work environment. During social development, individuals learn about roles, expectations that someone in a specific social position should meet. Since a MAS is created by autonomous agents, with different capabilities and, potentially, pursuing different goals, it is necessary to have a mechanism of bringing these capabilities together and controlling the agents' autonomous nature. It is suggested that each agent needs to be assigned a role that defines its behaviour in the organisational context at a given time. Similar to human societies, the notion of role can be used to balance the autonomous nature of agents with the need to have a structured organisation with a common goal (Schillo et al., 2002). The same structure can be transferred to agent organisations where, by defining roles and assigning them to agents, groups of co-operating or even competing agents can be formed in order to accomplish specific tasks. As Burton places it “role playing limits competition and promotes, if not active co-operation, at least peaceable co-existence” (Burton, 1969).

Furthermore, roles allow for the organisation to be decoupled by the specific agents that form it. This makes the organisation more flexible and resilient to changes. Having clearly defined roles, makes it easier to replace a member in case it leaves. It also makes it easier to introduce new members to the structure and help them operate in their new environment. For example, in a

market environment, it is easier to replace a settling agent, knowing that this role requires specific capabilities and has certain responsibilities, than to replace a specific company.

The decoupling of the organisation from the participating agents, also provides the interesting characteristic that it makes the organisation independent of the agents' internal structure. This is interesting as, in open systems, it is often impossible to know the internal structure of each participating agent. It is easier for the designer of an open system to design roles in an organisational structure, than design the actions of individual specific agents. This decoupling is also interesting to an observer of the system. As Burton places it "role enables us to abstract certain normative or stereotypes aspects of behaviour from the full repertoire of an individual's actions" (Burton, 1969). Knowing the roles that exist in the organisation, an observer of the system is more likely to succeed in identifying common behaviours that exist in the system and belong to a role.

Moreover, roles make it easier for an observer of the system to follow expectations, to measure the performance of the system. A role specifies the behaviour expected by the role holder. If this behaviour is measurable in any form, e.g. number of goods produced or quality of communication, an observer of the system can assess the overall behaviour. In an organisation without specified roles, it is difficult to assess the members' performance and how this performance affects the overall organisation's goal.

Role playing is central in understanding and designing organisational structures. Role is one of the attributes that characterise an agent organisation (Parunak and Odell, 2001). As seen in the discussion on organisational structure in section 1.2.1 above, the descriptive framework suggested in this thesis uses the notion of role as the building block of such a structure.

## 3.2 Existing definitions of role

It is accepted that, due to the reasons discussed above, roles play an important part in human organisations and, hence, in artificial organisations (Hörning and Lesser, 2005). However, a commonly accepted definition of what a role is does not exist. Researchers who use the notion of role as the building block of multi agent organisations provide different definitions of what a role is, how a role is specified and how it can be used in modelling MASs.

According to Giddens and Sutton (2010), a role has been defined in a sociological context as a "set of behaviours that should be enacted" by the person who occupies the role. Parunak and Odell (2001), define the role as a label on a

pattern of dependencies and interactions. They suggest that action dependencies and communication protocols are enough to define a role, emphasising the fact that the internal state of the agent playing this role is not important. A role for Odell, is a “class that defines a normative behavioural repertoire of an agent” (Odell et al., 2003a). Some researchers define roles as a set of prototypical functions, others as a set of obligations and permissions relating to the role (Sichman et al., 2005). Steegmans et al. (2005) see the role as the agent’s functionality in the organisational context and others as a set of objectives, rights and normative rules (Vazquez-Salceda et al., 2005).

In OperA, roles represent the different entities or activities that are needed to fulfil the purpose or the organisation, and are defined as a set of goals and actions Dignum et al. (2002a). In the Agent-Group-Role (AGR) model, a role represents the functional position of an agent in a group. A role describes the constraints that an agent needs to satisfy for enacting the role, the benefits that the enacting agent will receive from the role, and the responsibilities associated with it (Ferber et al., 2003). The Extended Gaia model defines roles in terms of responsibilities, activities and interactions with other roles (Zambonelli et al., 2003). Electronic Institutions define the roles as standardised patterns of behaviour that must be enacted by agents playing these roles Sierra et al. (2004), D’Inverno et al. (2012). In the OMNI (Organizational Model for Normative Institutions) model the definition of a role consists of an identifier, a set of role objectives, possibly sets of sub-objectives per objective, a set of role rights, a set of norms and the type of role (Dignum et al., 2005). Finally, the MOISE+ (extension of the Model of Organization for multi-agent SystEms) model describes a role as a set of missions assigned to the role in the functional aspect, and the set of obligations and permissions for this role in the deontic aspect (Hübner et al., 2002).

### 3.3 Characteristics a role definition needs to have

Despite the efforts to incorporate organisational aspects to agent oriented methodologies, none of these methodologies has been accepted as a standard (Isern et al., 2011). Seeing that different methodologies have their own interpretation and definition of the notion of role, it is evident that a widely accepted definition of the notion of role does not exist in this context. Therefore, as the role is considered one of the building blocks of the organisation, one needs to devise a definition that will provide the stepping stone to design MASs that incorporate organisational aspects and can be applied to open, heterogeneous, competitive,

self-organising systems.

This definition needs to meet certain criteria to make it adequate to be used in describing this type of MASs. It firstly needs to take into account the characteristics of agency. Agents are autonomous entities that can reason for themselves and make decisions according to their input. Roles are designed to tame the agents' autonomous nature, but they need to allow for a degree of autonomy. Furthermore, social ability is a defining characteristic of agents and one of the building blocks of an organisation. Roles need to allow for interaction between the members of the system and also allow for the definition of the communication patterns within them. Finally, another characteristic of agents is that they can adapt to their environment; the roles that specify the agents' behaviour need to allow for this adaptivity.

The definition of the notion of role should also match the characteristics that an open self-organising system has (as seen in section 1.2). In an open system, agents can join and leave the system at any time. Due to the open nature of the system it is not possible to know the internal architecture of all participating agents. As a result, the agents' behaviour cannot be predicted a priori. This means that roles should be de-coupled from particular agents (D'Inverno et al., 2012). In addition, the members of the system do not necessarily share a common goal. The organisational structure is imposed on members of the system to allow for cooperation and coexistence. This means that the way roles are defined needs to incorporate social control. As seen above, a self-organising system needs to be capable of adapting its organisational structure, and ideally following a bottom-up approach, without the need for central control. If roles are not needed anymore they should be removed, and if new roles are needed they should be created or existing roles should be altered. The definition of role needs to allow for evolving roles.

Finally, another characteristic is that this definition should include the notions that constitute a role in a way that can be easily understood and used in the design of computer systems.

To summarise, the definition of role to be used for designing open self-organising systems, needs to have the following characteristics:

1. The role needs to allow for a degree of autonomy. They need to be specified in such a way that the agents' behaviour is allowed to evolve within the role and to evolve the roles themselves. It needs to allow for new roles to be created, existing roles to be altered, or even discontinued.
2. The definition of role needs to include the communicational aspect of the agents' behaviour.



3. Roles need to be decoupled from specific agents and from agent specific internal mechanisms.
4. They need to allow for, and help, the maintenance of social order.
5. This definition needs to be expressed in a way that can be applied in a programmatic way.

### 3.3.1 Prominent models and Roles

This subsection reviews some of the prominent AOSE models, identified and reviewed in section 1.5, with regards to the characteristics identified above. The models that were identified as allowing for role-reassignment and role-evolution are discussed below.

It is noted that the ADELFE model aims to design self-adaptive systems, but it does not use the notion of role Bernon et al. (2004) and is not reviewed below. Also, INGENIAS does not suggest a definition for the notion of role as the rest of the models discussed in this section do. Roles are loosely linked to functionality, but the model does not provide a definition to be analysed here.

In **O-MaSE**, a role defines a position within an organisation whose behaviour is expected to achieve a particular goal or set of goals DeLoach and Garcia-Ojeda (2014). Roles are specified through an iterative design process, where the initial goals model, and the initial roles model, produce a detailed description of goals which are then assigned to a final set of roles. This definition of role, does not explicitly limit the agents' autonomy to act outside their role. However, at the low level design step, the designer needs to define the exact agent behaviour using an Agent State Model DeLoach (2005). In addition, the roles are completely decoupled by the role-holders. A role can be played by any agent, as long as they have a specific set of capabilities.

The communication model of O-MaSE is not part of the role definition, but is linked to roles by defining interaction protocols between roles and between the system and external actors. Moreover, O-MaSE's definition of role does not explicitly deal with social order. Order is maintained by specific agents, Organisational Agents (OAs), that capture the organisational hierarchy. Finally, with regards to the applicability of roles, (DeLoach and Garcia-Ojeda, 2014) offer a definition and an analysis process that allows for expressing roles in a programmatic way through the notion of goals.

The notion of role is important in **Electronic Institutions (EIs)**. Roles define patterns of behaviour within the institution, and each agent which participates in it must occupy at least one role Arcos et al. (2005). Roles are defined

while the dialogical framework is defined and can be of two types; internal or external. Internal roles are played by agents that belong to the institution, while external are played by agents that interact with it. There have been successful implementations of EIs, proving that the definition of role is programmable.

Roles in EIs allow for a degree of autonomy for the participating agents. Agents can switch roles, and behave as they wish according to specified scenes and transitioning between them. However, the role holding agents are not allowed to evolve the roles themselves. Communication is an important part of the role definition, as the roles are part of the dialogical framework. The dialogical framework also allows for decoupling agents from roles, with the added benefit that it caters for heterogeneous agents. Finally, the way to maintain social order is not built in the roles for EIs. Social order is defined in the set of normative rules, which are observed and enforced by the system's infrastructure.

The model suggested in **D'Inverno et al. (2012)** (mentioned as Communicating Open Systems - COS in table 3.1) uses a similar definition of role to EIs. Roles are defined as standardised patterns of behaviour that agents must respect when enacting a role. The model requires every agent in the organisation to adopt a role, and roles are completely decoupled by the enactors. As EIs, agents are allowed to transition between scenes and to change roles, but there is no provision for evolving roles.

Communication is not a core part of the role definition, and communication patterns between roles are designed through the Interaction mechanism. The authors also suggest that the mechanism of maintaining social order is part of the system's infrastructure, and is not directly included in the roles defined. Finally, the definition provided allows for roles to be designed in detail and in an unambiguous, programmatic, way.

The **OperA** framework describes roles in terms of requirements for interaction, communication, behaviour and interfaces. It decouples the actual roles, which are defined in the organisational model, and the agents playing these roles, which are defined in the social model. As with the EIs, the definition of role offers partial autonomy, as agents are allowed to assume different roles between scenes, but there is no provision for the agents' behaviour to evolve the roles.

The definition of role OperA offers partly covers the social control mechanism as well. The definition does not provide for designing the social order mechanism, but roles can be classified as social and operational. Social, or facilitation, roles are played by mutually trusted agents and aim to maintain the order of the society Dignum and Weigand (2003). The level of abstraction that

roles are defined on, shows that roles can be applied programmatically.

In addition to the four models discussed above, three further models (Extended Gaia, OMNI, and MOISE+) provide interesting characteristics with regards to roles and are discussed below.

The **Extended Gaia** model (Zambonelli et al., 2003) does not allow for autonomy from the agents' part. After the design phase, where specific agents are defined in order to play specific roles, it is not possible for these agents' behaviour or for their roles to evolve. With regards to communication, it is part of the role definition since it is defined as interactions associated with a role. The model allows for participating agents to be decoupled from specific roles. During the analysis phase the basic skills of the organisation are defined. The skills are elicited in responsibilities that each role has and it is at that point that agents are assigned to roles. Social order is not seen as part of an agent's role. The authors state that the social constraints can "hardly be expressed in terms of individual roles or individual interaction protocols". Organisational rules are used for defining social control. Finally, the model does not deal with implementation issues. However, it produces a detailed specification that can be used to build MASs.

Autonomy is achieved in the **OMNI** model (Dignum et al., 2005) by separating the abstract organisational level and the implementation level. Doing so, the model respects the agents' autonomy, while ensuring that they conform to the organisation's values. Communication in OMNI is defined in the interaction structure. It is provided as a set of scenes that follow a pre-defined script. These scripts are realised by joint activities between role holders, making these specific communication patterns an objective of roles. In addition, OMNI presupposes that the participating agents are designed separately from the organisational structure, meeting the characteristic of decoupling between the agents and the roles they hold. With regards to social order, values are defined to describe what is the expected outcome at the society level. Abstract Norms are defined which, when fulfilled, they contribute to a value. Through an iterative process, concrete norms that are translated into actions and concepts are designed. Finally, these more concrete norms are translated into rules, violations and sanctions that enforce them. Separating the abstract level and the implementation level, OMNI allows for interpreting abstract norms into more concrete ones. An iterative process is suggested to accomplish this. However, the model does not suggest how complex abstract notions can be translated into purely concrete norms.

Roles are not fully adaptable in the **MOISE+** model (Hübner et al., 2002).

However, the extension S-Moise+ provides a special role that is in charge of reorganising how different tasks are assigned to agents Rodriguez-Aguilar et al. (2015). Each role contains certain missions in the functional aspect. At the deontic level, a set of obligations and permissions describe which missions a role-holder ought to or is allowed to commit to. Communication is defined in terms of links, relations between roles. A link can be of an acquaintance, communication, or authority type. Roles are decoupled from specific agents; the model does not presuppose that agents can perform certain tasks. It does assume, though, that a role-holder will have the ability to perform the missions associated with this role. Furthermore, the definition of role, and the model itself, do not prescribe how agents that do not follow the organisational norms would be dealt with. The MOISE+ model can be expressed in a programmatic way having separate structural and functional aspects, which can be translated into missions and permissions/obligations to these missions.

The table below summarises the discussion above as to whether these commonly used models meet the characteristics described above.

Table 3.1: Role definition characteristics and models

Characteristic	O-MaSE	EII	COS	OperA	Extended Gaia	OMNI	Moise+
<b>Autonomy</b>	Partly	Partly	Partly	Partly	Partly	Yes	No
<b>Communication</b>	No	Yes	No	Yes	Yes	No	No
<b>Decoupled</b>	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<b>Social order</b>	No	No	No	Partly	No	No	No
<b>Application</b>	Yes	Yes	Yes	Yes	Yes	Yes	Yes

### 3.4 Behaviouristic and Mentalistic approaches to roles

Despite the fact that there is no commonly agreed definition for what a role in this context is, two tendencies can be distinguished in the definitions provided by various researchers in the field of Multi Agent Systems. Some define this notion based on the internal states of the agents occupying a role and others try to follow a behaviouristic approach defining the behaviour that role holders need to follow.

The former specify roles in relation to the beliefs, desires and intentions of the agents or other notions that relate to internal, or mental, states of an agent.

These definitions contain descriptive sentences that define the conditions for an agent to occupy a role and normative sentences applying to agents-holders of this role.

For example, Panzarasa, Norman, and Jennings ((Panzarasa et al., 1999)) see the role as “a system of prescribed mental attitudes, rather than a system of prescribed behaviour” (p.1). In their model, the role provides the agent with the mental attitudes, such as information on how to reason and on how to achieve its goals. These attitudes drive the agent’s behaviour as a result. A commonly used model for designing agents is the Belief Desire Intention (BDI) model. This model is based on the BDI logic developed by Rao and Georgeff (1991) and specifies roles and the agents’ behaviour based on their representation of the environment (beliefs), their high-level goals (desires) and their plan of action in specific cases (intentions). As a software development architecture method, BDI sees agents as rational entities that have certain mental attitudes of belief, desire and intention. These mental attitudes determine the system’s behaviour and are critical for achieving optimal, or the best possible, system performance (Rao and Georgeff, 1995). The BDI model has been and is being broadly used by researchers in order to design and build working rational agents.

A point of critique on the initial BDI models was that they do not specify mechanisms for interaction between agents and for building multi agent systems (Georgeff et al., 1998). The purpose of MASs is to take advantage of the agents’ combined behaviours in order to achieve a greater goal, which cannot always be achieved by a single agent. One’s behaviour in an organisational structure can only be seen in conjunction with other roles’ behaviour. Social ability is, hence, a crucial characteristic. Several BDI models, such as the InteRRaP and TouringMachines have extended the model and cater for the social behaviour of agents at the architectural level.

Another point of critique, which is common for the mentalistic approaches, is that mental notions are not easy to be translated into programmable logic. Consequently, it is very difficult to grasp their meaning in software development. BDI models use anthropocentric notions such as mental states and actions. Despite the fact that a number of BDI models has been developed, there is a gap between the organisational theory behind them and their implementation (Móra et al., 1999). Even when these notions are defined using some form of logic, this formalisation does not have an adequate operational model to support it.

It is this step that is the most difficult and crucial from a development point of view. The developer of such a model needs to translate notions such as belief,

desire, intention, deceit and power into software. The fact that they are not rigorously and unambiguously defined, leaves the details open to the developer's interpretation. Dealing with open systems, where agents are possibly developed by different teams, there is a gap between the prescribed mental attitudes and how these are represented into each agent.

The behaviouristic approach aims to define roles based on the behaviour expected from the agents. For example, Odell et al. (2003a) suggest that a role is "a class that defines a normative behavioural repertoire of an agent" (p. 3). A role, in their work, is a label to a set of dependencies and interactions with other roles. In another examples, Steegmans et al. (2005) design the roles as a diagram of interdependent behaviours, and D'Inverno et al. (2012) define roles as standardised patterns of behaviour.

Furthermore, Parunak and Odell (2001), following a behaviouristic approach, specify roles as a set of dependencies and interactions. From a systems analysis point of view this is beneficial since it is common practice to define dependencies and forms of interaction between different components.

In addition, designing a system without taking into account the components' internal states provides the opportunity to handle different entities, from agents to groups, as 'black boxes' that behave according to a pre-specified way without the need to analyse how they reach this behaviour. This characteristic allows for specifying more complex and scalable systems offering the characteristic of encapsulation. Encapsulation is one of the main service design principles of modern software architecture techniques, e.g. of Service Oriented Architecture. Encapsulation states that services, or agents, should not expose any details of their implementation (IBM, 2007).

From the analysis above it is clear that there are no right and wrong definitions of role. Some definitions are adequate for some models, and others meet more of the criteria specified in the previous section. This thesis is not aiming to provide a widely accepted definition for the notion of role. It does, however, provide a definition that meets the criteria set and is adequate to be used in the design of open, heterogeneous, competitive, self-organising MASs.

### 3.5 Definition of role

As seen above, several definitions for the notion of role have been introduced by researchers in different methodologies and different representations of role have been used in different models. These definitions vary based on the scope of the system designed, the theoretical model used and the design of the system's

organisational structure.

This thesis suggests that the behaviour of agents in open self-organising multi agent systems can and should be defined without using references to internal mental states. Borrowing the ideas of the doctrine of radical behaviourism one can argue that an agent's behaviour can be described without making references to the agent's internal mental states. A role, seen as a set of expected behaviours from the part of the role holder, will need to be defined strictly by defining the holder's behaviour in response to environmental stimuli. "The objection to inner states is not that they do not exist, but that they are not relevant in functional analysis" (Skinner, 1953).

By functional analysis Skinner means the analysis to identify the causes of a specific effect. In particular, if one needs to predict or control the behaviour, or function, of an organism, they need to identify and analyse the external variables that cause this function to happen. Skinner states that internal mental states play their role in the chain of events that lead to a behaviour. However, these are irrelevant to a scientific analysis that is trying to determine what causes one's behaviour and how this behaviour is controlled in an organisation.

This approach is appealing in designing an open system as it is impractical to be able to know each agent's internal configuration and how its internal states influence its behaviour. The analysis and the design of the system need to be able to bypass this stage and only take into account the agents' behaviour. As discussed above, this provides the characteristics of encapsulation and decoupling. Also, it bypasses the task of designing and implementing the logic behind mental notions; a task which is both difficult and sometimes reliant on subjective views. For these reasons, the roles of the agents' organisation need to be defined in behavioural terms.

Based on the above, this thesis suggests the following descriptive definition of the notion of role:

A role is an open set of stimulus-behaviour pairs. Where:

- Stimulus is a change to the local environment.  
A group of similar environmental stimuli is denoted as  $S_i$ .
- Behaviour is defined as an action, or re-action to environmental stimuli.  
A group of similar behaviours is denoted as  $B_i$ .

For simplicity in the discussion below,  $S$  and  $B$  will denote groups of stimuli and behaviours accordingly unless specified otherwise.

If we denote a group of stimuli as  $S$  and a group of behaviours as  $B$ , a pair can be defined as  $P: \{S, B\}$ . A role, will then be defined as the set:  $\{P_1, P_2,$

...,  $P_n$  }.

With regards to the point above, about groups of Stimuli and Behaviours, it needs to be noted that a step-by-step process of defining the groups of stimuli and behaviours, and which fall into each group, is out of scope of this thesis. This is a process that needs to be initiated by the system designer who, using their best knowledge of the environment and of the participating agents, needs to produce an initial grouping of predicted environmental stimuli and behaviours. As the system evolves through role alteration, new environmental stimuli and behaviours are either assigned to existing groups or new groups are created.

It also needs to be noted that a role is not only defined by the behaviour of the members based on the environmental stimuli, it is also defined through the role's interaction with other roles. Without interaction with other roles, a role is not of much use in an organisational structure. In the definition suggested by this thesis, role interaction is achieved with the use of stimuli being produced by a member's behaviour. Communication is discussed in detail in chapter 4.

In the definition above, and with regards to building roles into an organisational structure, the following also apply:

- Each agent must occupy a role.
- Each agent can occupy more than one roles.
- Each role can exist in more than one instances at a local level if needed.
- A role, of which only one instance exists at the local level and is not occupied by any agents is considered an 'orphan' role.
- A role can contain an infinite set of pairs 'P'.  
In real-life applications this is not the case. The highest allowed number of sets included in a role will be restricted based on the specific system's physical constraints.
- The set is not 'closed'; it can be updated with new pairs of the type {S,B}, or pairs can be removed.
- The behaviour of the role holder does not need to change if the environment changes; an environmental stimulus does not always need to cause a different behaviour. The same behaviour can be grouped with more than one stimuli.
- For each group of stimuli S, there can be more than one paired groups of behaviour B. This would lead to different pairs {S,B}



It needs to be noted again that this thesis also defines communication, one of the important aspects of an organisation, as a set of behaviours. Behaviour based communications are discussed in chapter 4. The next section describes how role assignment is defined and performed based on the definition of role discussed in this section. The subsequent section discusses the process of creating new roles and of the evolution of roles.

### 3.6 Assignment of role

Roles in an organisation are not permanently linked to specific members. At each given point in time, if one takes a snapshot of the organisational structure, each member/agent holds a role. However, this structure is bound to change as the environment changes, new agents might join the system, and others might leave.

Agents might become unavailable or leave the system at any time. In this case, the role that is occupied by these agents will be classified as an orphan role. This orphan role will need to be re-assigned to other agents.

In addition, an agent's capabilities might change over time. A software upgrade, a planned event or, simply, the deterioration of an agent's service might lead to alteration of its behaviour. This means that, in some instances, specific agents will not be able to perform the behaviours specified by the roles they hold.

Furthermore, if a role itself changes, a different agent, more capable of fulfilling the role, might need to be assigned to it. Roles are not static entities in the organisational structure. Each role can be enhanced, with more {S,B} pairs added to it, or simplified if some {S,B} pairs are not needed anymore for the specific role. In this case, the agent assigned to the role might not be able to fulfil this role, or might be overqualified for it.

Finally, during the system's lifetime, new roles can be created if needed. These new roles will, initially, be orphan roles that will need to be assigned to qualifying agents.

(Derakhshan et al., 2013) define the following sources of dynamism in open MASs:

1. Changes in the population of agents as they join and leave the system.
2. The results of actions performed by the agents.
3. Changes in the environment.

4. Important thresholds being reached, for example the end time in an auction.

A mechanism for assigning roles to agents is introduced to the framework to ensure continuity of the organisational structure. A process of roles' assignment will need to be used both when the system is initiated and during runtime. One can categorise the role assignment processes as exogenous or endogenous (Odell et al., 2003a). Exogenous role assignment refers to the process where an external source, such as the system designer, assigns roles to agents, while endogenous assignment is the process of self re-organisation where the agent organisation itself re-assigns roles to participating agents.

When the system is initiated, the roles need to be assigned exogenously. The designer, having knowledge of the initial status, decides which agents will occupy which roles. At this stage, the system designer uses their knowledge about the agents' capabilities in order to create the best possible match between the roles that are needed to form the organisational structure and the agents available at this time. This is useful as it gives an initial guidance to the system, reducing the need for immediate self-organisation. Also, if the system designer assigns roles that are close to a human-like organisational structure, it leads any subsequent endogenous re-assignment to stay close to this structure.

Over time, due to the open nature of the system, many factors can influence and change this initial assignment of roles. This re-assignment of roles can either be done with the designer's intervention, exogenously, or without it, endogenously. In designing systems that are prone to frequent or unexpected changes, it is considered better to have a mechanism for endogenous re-organisation (Odell et al., 2003a). This design makes the system more robust and utilises less human resources over time as the designer's intervention is not required every time a change occurs. Endogenous role assignment is common in societies of social insects (Beshers and Fewell, 2001).

In the secondary market sector, for example, it is common for regulated markets to have the concept of a market maker. The market maker is a participant who always provides a quote to buy and sell specific stocks. This way it is ensured that there is liquidity; every market participant that needs to create a position in a stock has someone to trade with. If at a certain point in time a market maker becomes unavailable, for example because they fail to hold positions in certain stocks, this role needs to be assigned to a different participant. Some markets have a structured approach where the market itself decides which member can start market making. In other markets, a more liberal approach is followed where every member that has the capacity can start market making.

Due to the different reasons that might cause the system to re-organise, the descriptive framework proposed in this thesis suggests a mechanism for endogenous role assignment which describes when a role needs to be re-assigned, how this is identified, and how it is performed.

### **3.6.1 When is role re-assignment needed**

A role will need to be assigned or re-assigned endogenously if at least one of the following conditions apply:

- An agent becomes unavailable.

At each point in time an agent might become unavailable. This might occur due to the agent losing its communication link with other agents or due to the agent failing and stop functioning. This can be either caused by changes in the environment or by the agent itself. In any case, this will have as an effect that, from the other agents' point of view, this particular agent is not performing the role anymore.

- An agent joins the system

When an agent joins the system, a role needs to be assigned to it so it becomes a member of the organisation.

- An agent can no longer hold a specific role.

Over time, as the organisational structure of the system changes, an agent which is assigned to a role might not be able to perform it anymore. This could happen either because the role has changed and is beyond the agent's capabilities or because the agent's capabilities changed so that it can no longer perform certain functions required by the role. For example, an agent might receive a software update that removes specific functionality from its repertoire, or one of its sensors might fail. If this causes the agent to stop being able to perform certain functions of the role, then the role would need to be re-assigned.

- A new role or role instance is created.

If a new role is created, or if a new instance of an existing role is needed at a local level, then this role needs to be assigned to an agent.

### **3.6.2 How to identify that role re-assignment is needed**

If one of the conditions above occur, then the system, at a local level, needs to understand that a role instance needs to be assigned to an agent. If an agent

fails or is not able to perform a role anymore, there are two possible cases; either the other agents need to identify this change in behaviour, or the agent can let the other agents know that a role re-assignment needs to take place.

Since a role is also defined through its interaction with other roles, the fact that a role is orphan will be identified by the agents that hold roles interacting with it. A role contains specific pairs of  $\{S, B\}$ . The fact that the role-holder ceases to play this role means that certain stimuli, which are the effect of the role-holder's behaviour, will cease to exist. When this happens, the agents of the local environment need to identify if any other instances of this role exist. If this is not true, or these instances do not produce the desired behaviour, the orphaned role instance will need to be re-assigned.

The second case is preferable as it gives other agents time to prepare for this event. In this case, the agent that is due to release itself from a role needs to communicate to the other agents of its local level that it will no longer be holding the specific role. However, this is an ideal scenario that is implementation specific and cannot always occur. An agent might not have a mechanism of identifying its status, so if the agent is going to fail it might not be able to identify this. Also, even if the agent is able to realise and communicate its status, other members might not be able to receive or understand this communication. Furthermore, an agent might fail abruptly; this would not give enough time, even to an agent with self health checks, to notify the other agents of the organisation. In this case, the agents that hold roles that interact with this particular role will need to identify that this event has occurred.

Furthermore, the second case assumes several implementation details of the agents; that they can directly communicate with other agents using predefined speech acts, that the failing agent has this specific message in their repertoire and that the receiving agents can understand this message. As seen above, this model does not presuppose anything about the agents' internal architecture. However, this ideal scenario is not dismissed. It can be potentially seen as a subset of the  $\{S, B\}$  pairs where the message is a particular group of stimuli and B is the relevant group of behaviours.

The above also stand in case a new agent joins the system. The other local agents can either see a change in the local behaviour or the new agent can explicitly declare its presence. A change in behaviour will be seen simply because the new agent will start acting/reacting to changes in the environment. These behaviours will produce stimuli which will, in turn, trigger a behaviour from the part of the existing agents. As above, the special case where the new agent presents itself through a predefined message, is seen as a subset of the

{S,B} pairs.

### 3.6.3 How is role re-assignment performed

After it has been identified that a role needs to be assigned to an agent, through the existence or absence of specific stimuli, the mechanism of endogenous role assignment needs to be triggered. As discussed above, role re-assignment can be performed either endogenously by the system itself or exogenously by an external factor, such as a system administrator. For resilience, better use of resources and for avoiding bottlenecks, it is preferable that role re-assignment is performed endogenously.

There are various different mechanisms of endogenous role assignment. For example, an agent with specific capabilities can have the role of the decision maker. When a role becomes orphan, the agent can assign this role to agents that express interest in it. In BRAINS (Behavioural Roles for Agent INteractions) for example, role assignment is performed endogenously (Cabri et al., 2003). However, it requires agents with special capabilities (the role loader) as well as a central repository of role descriptions to exist. Both these prerequisites introduce a bottleneck and a single point of failure to an open system.

Alternatively, agents that express interest to an orphan role can bid for it. In this case the best bidder would get the role through a specific bidding process. This scenario would also require the agents to have knowledge of the bidding process and its rules. Moreover, this would require every agent joining the system to be familiar with this bidding process.

Another suggested mechanism is for a separate component of the system, i.e. not the agents themselves, to make this decision. This is the case in d’Inverno’s methodology where the system’s infrastructure is responsible for assigning roles to participating members (D’Inverno et al., 2012). This approach also introduces a bottleneck and a single point of failure; if the infrastructure fails, the agents will not be able to continue this task.

This thesis suggests a model influenced by the response-threshold model used to describe social insect organisations (Robinson and Page, 1989). This model tries to describe how division of labour is performed in social insects’ societies. It tries to answer the question why, in bee hives for example, some bees forage while other bees stay at the hive. And, more importantly, what makes specific bees to assume the role of a forager. How does this polyethism come to life from the individual members’ behaviours.

The response threshold model is built based on the hypothesis that workers have different thresholds to reacting to different stimuli. This variation in

thresholds is what generates labour division as different members assume different roles. The model also contains a negative feedback loop in which the behaviour of the worker with the high threshold reduces the stimulus (Beshers and Fewell, 2001). As Beshers and Fewell place it “variation in response thresholds generates a system that combines individual task specialization and colony task flexibility.”

Polyethism has been observed to be of two types, temporal and morphological (Beshers and Fewell, 2001). Temporal polyethism refers to the division of labour based on maturity, on age, or experience. For example in bee hives, it is the elder bees that assume the role of the forager and the younger bees stay at the hive. Morphological polyethism describes the division of labour based on physical characteristics. Larger ants for example, will assume the role of defending the nest. Temporal and morphological polyethism give an interesting distinction which could be used in the design of open MASs as future work. Temporal can refer to a queue like system where the more established agents can select their roles, or have more freedom to introduce new behaviours. Morphological polyethism could be used to describe the process where agents with different capabilities and skills assume different roles. This distinction is left as future work for this thesis.

The threshold response model can be adapted for use in the model suggested by this thesis due to the openness of the systems involved. Each agent has a repertoire of actions, with each action being invoked by an environmental stimulus. Given the different capabilities of the participating agents in an open system, as well as their different decision making mechanisms, it is fair to assume that not every agent will have the same response to the same stimulus. Also, even if two distinct agents are designed to have the same behaviour, it is not certain that this behaviour will be invoked at the same time; an agent might have better sensors, might have a faster decision making process, or might not be that interested in a particular stimulus. Since each role is defined as a set of {S,B} pairs, the response threshold model states that not all agents will:

1. Be interested in the stimuli at the level they exist, as their threshold might be lower.
2. Be able to react to the stimuli available in the role.

As a result, an orphan role will be occupied by the agent that first reacts to the stimuli contained in this role. Similar to a bee that has a higher threshold of hunger and, hence does not react to the stimulus of hunger by foraging, agents that are not interested in the orphan role, or do not have the ability to react to

its stimuli, will not try to occupy it. Also, similar to when two bees that react to the same stimulus, the one with lower threshold will try to perform a task, agents that have more need to occupy a role will do so.

The negative feedback loop suggested by the model is also useful in the MAS paradigm. The negative feedback loop ensures that the stimulus is reduced when the relevant task is performed. If the bee with lower hunger threshold starts foraging, it will bring food back to the hive, allowing the other bee to feed itself. This ensures that division of labour remains as long as it is needed, allowing the second bee to perform a different task. In a similar manner, the response threshold model allows for agents that are 'in need' of a role to occupy that role, freeing the other agents from this behaviour allowing them to perform other tasks.

Polyethism can be observed in the secondary equities trading market in certain scenarios. The most obvious example is when the need for liquidity arises for a particular stock. The term liquidity refers to how easy it is to find the opposite side of a trade to fulfil your orders.

In a common sequence of events:

1. A Fund decides that it needs to acquire a specific stock.
2. The Fund places an order with a Broker.
3. The Broker tries to fulfil the order on different sources:
  - On the regulated market where the stock was issued (e.g. on the London Stock Exchange).
  - On other markets where the stock is traded (e.g. on the Euronext exchanges and on the German exchange).
  - On deregulated markets, such as Crossing Networks and Dark Pools.
  - By contacting other market members known to trade this stock.

If the stock is not traded actively in any of the above places, and especially on the regulated markets, then the stock is considered illiquid. In order to provide liquidity, markets have introduced the concept of market makers, and of liquidity providers. A liquidity provider is a firm that is known to always provide the opposite side of an order in a regulated market for a specific set of stocks.

In the scenario above, where the Broker could not fulfil the order due to the stock being illiquid, the following could happen:

1. A Broker that has an order on the same stock for the opposite side can provide liquidity.
2. A Broker who usually deals on this stock, or specialises on this market, can start providing liquidity.
3. A Broker that has a strong balance sheet, can take the risk and start providing liquidity on this stock.

The first two cases above are an example of temporal polyethism. The Stimulus in this example is the fact that a stock became illiquid, and the Behaviour is to start providing liquidity. Moreover, this Pair is part of a Liquidity Provider role.

The brokers that specialise in this market, have a lower threshold for the Stimulus (possibly because they are interested in that market performing well) and started assuming the role of the liquidity provider.

The third case shows morphological polyethism. The third broker does not necessarily have a lower threshold to the Stimulus, but it has the capability to assume this role.

### 3.7 Alteration of role

As seen in section 3.6, a role might need to be re-assigned to different agents during time. Due to the changes in the environment and to the fact that new agents might join the system or agents might leave the system, a role might also need to be altered. Self-organisation is fully achieved through role alteration, role creation or role deletion when specific circumstances emerge.

Based on the definition of role given above, role alteration happens when the set of pairs  $\{S, B\}$  changes for a given role. One or more sets can be added or several sets can be removed. Role creation can be seen as a more radical form of role alteration. A role is created by a blank set of pairs and then it is defined by adding new  $\{S, B\}$  pairs. Finally, roles can also be removed at a local level. If a role is no longer needed, it can be removed from the group of roles in the specific local environment.

As with role-assignment, roles can be altered either endogenously or exogenously. Exogenous role alteration would require a user of the system, for example the system administrator, to add or remove pairs of  $\{S, B\}$  for a given role. This exogenous operation would be needed at the initialisation process of the system. Endogenous role alteration will be required during time. As the system's environment evolves, new environmental stimuli might emerge. These



stimuli would cause the agents to produce different behaviours, adding a new pair of  $\{S,B\}$  to their repertoire. Furthermore, a new agent joining the system might introduce new pairs of  $\{S,B\}$ . For example, at a given time an environmental change might not be stimulating a behaviour to the existing agents of the system. However, a new agent with more capabilities might join the system adding new functionality and causing new types of behaviour. An agent with different sensors might be able to identify some changes that others were not able to identify previously. This will bring in a new pair of  $\{S,B\}$  to the system as well. Role alteration can also occur when existing agents start playing new roles. Each agent can, potentially, be different having a different repertoire of behaviours and various sensory abilities. As a result, each agent playing a role could actually influence the role itself by adding or removing  $\{S,B\}$  pairs to the role.

A very interesting example of this process is the way honey bees in the Aegean islands deal with strong winds. Usually hives are placed on hillsides, while plants usually grow by the sea. It has been observed that, during windy days and only during these days, beans of sand are on the landing part of the hive. Zoologists have observed that bees actually carry these beans of sand on their way back to their hives, to increase their wind resistance. These foraging bees have evolved their role in order to adapt to different environmental conditions.

As new sets of  $\{S,B\}$  are introduced to the group, either due to new stimuli or to new functionality, they will also have the potential to become part of the roles that the agents hold. However, not all new pairs will have this effect on a role. The following are possible when a new pair  $P_{new}:\{S,B\}$  is introduced by an agent. Note that the  $P_{new}$  can be the result of the new agent being able to recognise more stimuli ( $S_{new}$ ), the agent bringing new behaviour to the system ( $B_{new}$ ), or both.

1. The new pair produces a new behaviour ( $B_{new}$ ) to an existing stimulus.
2. The pair brings a new behaviour because a new stimulus is recognised.
3. The pair recognises a new stimulus but an existing behaviour is produced.

From the other local agents' point of view, items one and two above are the same; they experience a new behaviour, through new stimuli in the environment. This new pair needs to be looked from two angles. Firstly, if the new stimulus is useful to the agents at the local level. Secondly, if it is in line with the social control mechanism. Both questions, about usefulness and order, can be

answered by a reference to evolutionary fitness. The local agents are expected to react to the new stimulus trying to maximise their fitness levels.

Let's assume, in the secondary markets example, a new broker dealer that joins the market. The broker has a better network of analysts who identify that a particular company is ready to announce very good profits (Snew). This would normally lead the price of the stock to raise. The broker starts buying stock in order to sell it in a higher price when the official company results are announced (Bnew).

The rest of the market participants would start selling to the broker, not knowing about the good results about to be announced; this would drive the price up in a moderate rate. Some of the participants might have the ability to analyse the current prices, seeing that the price of the stock is being raised unexpectedly. These brokers might start buying the stock as well to take advantage of this situation (i.e. to increase their fitness). In addition, they might remove all their sell orders from the market, so as to keep the stock until the results are announced.

The new behaviour on behalf of the broker has proven to be both useful (providing liquidity to the market) and according to the social order of the market. This behaviour had a twofold effect. Firstly, some participants realised that the market is moving and behaved bullishly (having this pair in their role repertoire) acquiring stock. Secondly, other participants who acted in a more conservative way removed their sell orders from the market, bringing it in an equilibrium. From the above, it can be seen that this new pair (Pnew) produced a stimulus which is both useful at a local level and retained social order. This Pnew can become part of the role that the initial broker holds.

Differentiation brings different behaviours and also helps to incorporate the useful ones to the system. This is important in an open system where the participants are not known and are likely to be different in terms of how they reason and behave. This phenomenon has been analysed in human societies by Bales and Slater (1955) and Peter Burke. These sociologists state that "a process of role differentiation takes place along task and social-emotional dimensions in group interaction" (Burke, 1968).

With regards to the third point above, when a new behaviour is not produced by the new pair, it is not possible for the local members of the system to identify that a new pair (Pnew) has been introduced. As a result, this new pair will go unnoticed. Roles are defined only in relation to other roles, a role is not a stand alone entity, it exists within the organisation. Consequently, it will be impossible for this new pair to become part of the role.

Another part of role alteration is the removal of roles. When a role is not needed anymore, it can be removed from the local environment. This is a process that follows the same logic as the role alteration described above. When a Pair is not needed anymore, agents can stop performing the relevant behaviours. If this removal aids the local agents' fitness and is in accordance to the social order, then it can become permanent. This way, roles that are not needed will become leaner and, overtime, they will extinct. This is a common phenomenon in human societies. For example, before the introduction of electricity and gas in public spaces, the role of lamp operator existed. This role defined that special workers would need to light up the lamps on the street every night and put them out every morning. Overtime, with the introduction of electricity (new stimulus) these role holders stopped doing this task or they did it in a smaller level. Finally, as the removal of the task was beneficial to the agents' fitness (the operators could be used in other tasks) and did not disturb the social order (the electric lights were lit at night) this role was completely removed.

### 3.8 Summary and evaluation

The definition of role described and discussed in the section above fulfils the criteria described above for defining the notion of role to be used in open self-organising systems.

- **Allow for autonomy** The definition allows for and encourages the autonomous nature of participating agents. Role assignment is flexible and can be changed endogenously by the agents themselves. In addition, the definition allows for the roles themselves to evolve based on their holders if this evolution is beneficial to the other participants and does not disturb social order.
- **Include the communication aspect** Defining a role as a set of pairs of the form stimulus-behaviour allows for the communication aspect of the organisation to be built on this. Communication is performed via the environment (stimulus) and as a cause of agents' behaviours (be it speech acts or implicit communication acts). Communication in the model is discussed in detail in the next chapter.
- **Be decoupled from specific agents** The definition does not assume any special agents assigned to any specific roles. The processes of role assignment and alteration are agent agnostic.

- **Allow for and promote social order** Social order is about limiting the autonomy of the participants in order to follow specific rules and to achieve a system goal. The definition of role which deals with agents' behaviours on environmental stimuli provides a good basis for promoting social order in the system. This is discussed in detail in chapter 5.
- **Be applied in a programmatic way** As discussed above, the definition of role provided in this thesis purposely avoids any reference to mental notions, or internal states of agents. This is mainly to avoid the complexity of logically defining and programming such notions. The fact that roles are simple pairs of environmental stimuli and behaviours allows for the implementation of such a structure.

This definition will be used to discuss the communications model (chapter 4) and the social order (chapter 5) of the descriptive framework suggested in this thesis.

## Chapter 4

# Communication

### 4.1 Communication in organisations

In order to design an open system and, furthermore, apply an organisational structure on a Multi Agent System (MAS), one also needs to define how the agents of the system communicate. Communication and interaction are characteristics of agency (see for example, (Sims et al., 2003); (Odell, 2007a)). Agents do not act as isolated entities; they are capable of interacting with their environment and other agents. For example, communicating a need for resources or information on the current state of the system. They are ‘social’ entities, designed to form groups and to communicate in order to achieve their goals. Communication, hence, plays a crucial role in the design of MASs.

Furthermore, communication is one of the key aspects of an organisation (Parunak and Odell, 2001). For the members of the organisation to be able to achieve their common goal, to co-operate and inform each other about their environment, a model of communication needs to be developed. Roles are only useful if they are played within a group, through communication and interaction. This thesis suggests a framework for designing open, self-organising systems using self-adapting organisational structures. For this framework to be complete, one needs to define the model of communication to be used between the members of the system.

This chapter lays the principles for designing communications for this specific type of MASs using the suggested framework. Firstly, it discusses the characteristics that a communications model needs to have, or avoid having, in order to be used in an open self-organising MAS. After these characteristics are identified, a review of existing models is given noting their relationship with these

characteristics. In the subsequent section of this chapter, this thesis provides a definition of what a communication act is in this context. It discusses the relationship between the environment and communications and finally provides a model of communication to be used in open self-organising MASs.

## **4.2 Characteristics of a communication model**

For a communication model to be used in the design of open self-organising systems, it needs to promote the principles of the system and also ensure that it does not contradict with any of the characteristics the system is designed to have. This section provides a list of these characteristics, based on the characteristics that an open self-organising system has.

### **4.2.1 Decoupling**

As discussed in Chapter 3, the systems to be designed using the proposed framework need to follow an organisational structure realised by the roles that exist in it. The agents occupy roles that, combined together, try to accomplish the organisation's aim. Each agent can occupy one or more roles and the roles define the agents' behaviour. A similar approach needs to be adapted for the communications model of the organisation.

The communicative behaviour of the organisation's members needs to be defined by the role they occupy. The role defines the behaviour of an agent and, hence, the communicative acts that are generated by this behaviour. There is a bi-directional relationship between the organisation's roles and the communication model. Through time, communicative behaviour can evolve while the role itself evolves. In the case where an agent with different behavioural capabilities occupies a role, new communication signals can be created. In addition, an agent with different sensors might be able to understand different and more communication signals than the previous occupier of a role. The emergence of a new communication signal, can lead to new behaviours, evolving the roles of the organisation. The above lead us to define the first characteristic that the communication model to be used needs to have: it needs to be used to define communication between roles, not specific agents. Communication needs to be decoupled from specific agents.

### 4.2.2 Signals' creation

The communications model is designed for use in open systems. This means that one cannot know all the possible states of the environment a priori. In the case of a well-studied closed environment, it is possible for the designer to predict all the states of the environment. Having this knowledge, one can design the expected behaviour of the agents and the communication signals to be used. However, in an open environment, the designer is not able to predict all the possible states. As a result, they cannot specify all the communication signals that will be used by the agents.

Another reason that the communication signals to be used cannot be pre-defined is that the designer cannot know each agent's ability to understand environment states, or to communicate them to other agents. Even if all the states of the environment were known, the agents' ability to understand these states and the way they would behave is unknown in an open system.

As a result of the system characteristics above, it is not possible to define all the communication signals to be used in the system from the beginning. This is the second characteristic of the communications model; that it needs to allow for the creation of communication signals. This is a common characteristic of open systems that, can be argued, is seen in all real-life communication systems. Signalling systems evolve; they are not fixed, closed interaction structures (Skyrms, 2010).

### 4.2.3 Internal information processing mechanisms

Furthermore, it is not possible to know each agent's characteristics, capabilities and internal structure at each given time. As a result, it is not possible to know if an agent has internal mechanisms of understanding communication signals or if they have mental abilities to process signals in a certain way. In order to design a communications model to be used in an open system, this model needs to avoid using mental notions. The third characteristic that the system needs to have is that it has to be based on information exchange without taking into account the agents' internal mechanisms. As Skyrms says: "The place to start is not with a self-conscious mental theory of meaning, intention or common knowledge, but rather to focus on information. Signals transmit information, and it is the flow of information that makes all life possible." (Skyrms, 2010).

#### **4.2.4 Communication language**

The fact that the system is open also means that the designer of the system can not know the communicative capabilities of each agent. They can not know the communications language each agent can use, or if they are capable of using a language at all. This leads us to point out the fourth characteristic of the communications model: there need not be a common, pre-defined language or signalling convention. Agents might be 'talking' different languages or might not be able to use a known communication language at all.

#### **4.2.5 Communication channels**

In an open system it is not easy to predict, at a given time, how many and which agents form the system. It is equally difficult to predict which or how many agents will join or leave the system. The communication model used needs to also have this flexibility. It is not possible predict between which roles communication will take place, especially since roles might evolve during time.

In addition, since the environment of the system is not stable, the communication model cannot rely on dedicated or pre-defined communication channels between the agents.

This is the fifth characteristic that the communications model needs to meet. Communication signals are transferred through channels. However, these communication channels cannot be pre-defined. The nature of the environment can lead to new ways of communicating and make existing communication channels obsolete.

#### **4.2.6 Decentralised structure**

Another characteristic of a self-organising MAS of interest to this thesis is that there is no central control on the organisational structure. There are no overarching roles that determine the system's structure. The communications model needs to follow the same constraints. It needs to be independent of roles that control the communication between agents. In addition, the model needs to not rely on any special communicative characteristics that specific agents might have. For example, the model needs to avoid agents that centrally route and disseminate signals between other agents. This is a common practice between engineers that has proven very useful in existing applications. However, in a self-organising system this constitutes a single point of failure and needs to be avoided.



The sixth characteristic that the communication model needs to meet is that communication needs to not be performed in a centralised manner. This does not exclude the possibility of having a central communications agent at a given time with more responsibilities. It requires, though, that the communications model is not relying on this agent.

#### 4.2.7 Deceitful communication

Finally, the communications model used should allow for deceitful communication. Deceit is a common characteristic of communication models in real life organisations. Deceitful messages have been observed in societies of species of different cognitive ability (Skyrms, 2010). The model cannot presuppose that all communication in the system will be reliable. To only allow for reliable communication is seen as a limitation to a model, and also contradicts with the criteria of openness specified above. In addition, to assume that all the agents occupying roles in the organisation are only capable of reliable communication contradicts the criterion that agents of unknown design and behaviour are allowed to join the organisation.

To summarise, the characteristics of the systems to be designed using the framework suggested in this thesis also define the characteristics that the communications model needs to have. These are the following:

1. It is not agent specific and is used to define communication between roles, not agents.
2. The communication model needs to allow for the creation of communication signals.
3. It needs to be based on information exchange without presupposing given internal mechanisms of the agents.
4. It should not presuppose a common, pre-defined language or signalling convention.
5. The communication model cannot rely on dedicated or pre-defined communication channels between the agents.
6. Communication does not need to be performed in a centralised manner.
7. The communications model used should allow for deceitful communication.

### 4.3 Existing approaches

Several methods of communication between agents have been investigated and specified during the past years. The methods designed specify different layers of the communication models; from the way the communication channels are structured to specific communication languages and protocols. Most of the research on agent communications is focused either on the transport layer of the system, the session layer or on the presentation layer and high level design. The reference on layers is based on the OSI (Open Systems Interconnection) model of communications design, a widely used model for systems' design. In the OSI model, the transport layer describes the means of transferring data between different points of a network, the session layer describes sessions between applications and different hosts, whereas the presentation layer describes the way data and information is represented.

In MASs design, the majority of the suggested methods regarding the transport layer of the system is based on the existing network infrastructure and use the already established internet protocols or their variations. These methods include a peer-to-peer architecture based on unique agent IDs assigned to each agent participating in the system (Baumann et al., 1997) and a client/server model (Finin et al., 1994). Based on this set network infrastructure and internet protocols, researchers have designed several agent communication languages (ACLs), standardised languages for agent communication, and have built Agent Interaction Protocols (AIPs) based on these languages.

The discussion in this section and the model suggested are focused on the presentation layer. The other levels are seen either as implementation specific (e.g. which transfer protocol is to be used in a specific network), or are incorporated in the environment (e.g. the sessions, the available channels at a local level). The presentation layer in MAS design deals with what is communicated and how this is achieved. Several models define the universe of messages that can be exchanged by agents in each situation, the language to be used for this exchange, and what information is expected to be transmitted between agents.

Wajid and Mehandjiev (Wajid and Mehandjiev, 2013) provide a very interesting analysis of existing methods of communication for flexible, adaptable systems. They classify approaches about agent interaction as follows:

1. **Protocol based approaches.**

These are approaches where communication is defined based on a protocol that defines the messages to be exchanged in each scenario. These

approaches can be categorised further into the following.

- (a) Protocols are loaded at the design stage.

This approach is quite inflexible as it does not cater for any changes during runtime. All communication patterns need to be defined by the designer a priori. As a result, it does not allow for signal creation.

- (b) New protocols can be loaded at runtime.

With this approach, designers have the option to add new protocols in a pool of protocols allowing agents to use new protocols at runtime. This approach allows for signals' creation, but it does not meet other criteria. For example, it presupposes that every agent uses the same language. This method is also inflexible, it requires external intervention from the designer and in a centralised manner.

- (c) Flexible protocols, that the agents can create at runtime.

These flexible protocols can either be commitment based or based on joint intention. Commitment based protocols use agent commitments to help agents form the communication messages. Commitment based protocols have been quite successful in designing MASs, however there are a few drawbacks. Namely:

- These approaches make it more difficult to analyse different types of interactions, e.g. negotiation or persuasion cases.
- It is not clear how commitment can be defined, what does it mean for an agent to commit himself to an action. (Jones and Parent, 2007)
- They assume specific internal mechanisms of the agents, which is one of the characteristics specified above. Commitment based protocols assume that agents can understand the concept of committing to an action, and also that they are capable of creating communication patterns based on that.

## 2. Approaches without protocols.

These approaches define agent communication based on norms, on argumentation theory or on their beliefs and intentions. These approaches can be categorised into rationalistic and mentalistic.

- (a) Rationalistic approaches.

Rationalistic approaches, or commitment-based (Jones and Parent, 2007), can be argumentation based or based on dialogue games.

Argumentation approaches allow for agents to form conversations

as arguments. Based on an agent’s beliefs, a reasoning mechanism and a conclusion to be reached, agents can exchange messages and also reach an agreement or improve their knowledge. Argumentation based approaches are interesting, but they have the shortfall that they presume a high level of autonomy and intelligence on behalf of the participating agents. Agents need to be able to understand arguments, reason about them and respond accordingly.

Approaches based on dialogue games are similar. In dialogue games, agents act according to a predefined set of rules, the rule of the game. Dialogue based approaches offer the flexibility needed in open systems, as they only present the rules of the game, not the actual messages. However, as with the argumentation based approaches, dialogue games require a high level of intelligence on the part of the agents, requiring them to produce the messages to be exchanged on the fly, following the game’s rules.

(b) Mentalistic approaches.

Mentalistic approaches, or intention-based (Jones and Parent, 2007), are based on the ‘speech act theory’ Wooldridge (2009). As the most prominent methods fall into this category, mentalistic approaches are discussed in the remainder of this section.

Most methods of agent communication and most ACLs are inspired by the ‘speech act theory’ Wooldridge (2009). Another common characteristic of these methods is that communication is seen as intentional transmission of information about the mental states of the sender, open to the interpretation of the receiver.

Due to the large number of ACL specifications available, a group of researchers and members of the industry have created the Foundation for Intelligent Physical Agents (FIPA). FIPA’s aim is to “promote agent-based technology and the interoperability of its standards with other technologies” (FIPA, 2014). The foundation has produced a specification of an ACL to be used for implementing the communications aspect of MASs. This specification is driven by many researchers and is based on clear semantics (Kone et al., 2000). However, its semantics have limitations as they are based on the beliefs and intentions of the agents. As the FIPA Architecture Board mentions: “The meaning of the content of any ACL message is intended to be interpreted by the receiver of the message. This is particularly relevant for instance when referring to referential expressions, whose interpretation might be different for the sender and the receiver.”. This fact poses an issue; communication is based on the receiver agent’s ability to interpret the sender’s mental states.

Furthermore, another point of criticism is the fact that the belief or the intention of the agent cannot be verified by its actions. Since these are internal mental states, it is not possible for an observer to verify that the correct message was used for the situation (Jones and Parent, 2007). Due to this, characteristics of the communication model, for example the requirement to allow for deceitful communication, cannot be measured. As (Searcy and Nowicki, 2005) state "... 'intentions' and 'beliefs' are mental states, and as such are difficult to measure in non-human animals...". This applies to software applications as well.

Another widely used ACL is the Knowledge Query and Manipulation Language (KQML) (Finin et al., 1994). KQML is a message based language, based on speech acts, and is used to express agents' mental states. KQML has become broadly used in agents' communication, however it has not become the standard framework for communication. The main reasons for this are that the semantics of the language were never formally defined and that the language misses a class of commissive performatives which are necessary for transferring messages where an agent commits to perform an action Wooldridge (2009). In addition, in the KQML specification, the transport mechanism of messages is not rigorously defined. So, communication between two agents can only be achieved through a facilitator agent which takes the role to translate and disseminate the message to the relevant agents. This method offers interoperability as it allows for messages to be passed between different systems but it introduces a single-point of failure and a potential bottleneck to the communication process. If the facilitator agents fails, then the communication between the two systems becomes impossible. This issue becomes more apparent in the design of de-centralised MASs where indirect communication, without the need of a central overarching agent, is advantageous.

Other common ACLs are the ARCOL communication model (ARTIMIS COmmunication Language) and communication models based on mobile agent technology (for a review see (Kone et al., 2000)). The communication models described and mentioned above, as well as other similar models, have particular strengths but also certain limitations that do not allow them to be widely used in MASs' specification methods (Kone et al., 2000). The majority of these models see communication as the intentional transmission of information regarding an agent's mental states. For example, in KQML a communicative act is a message that expresses an agent's mental states such as its beliefs and intentions, introducing a degree of ambiguity to the message.

Finally, even though these methods of communication have the advantage that they are based on known and tried protocols and ideas, the agents are

expected to know how to use this infrastructure and the language of communication. This leads to 'low heterogeneity' and does not fit well with the view that MASs are open systems, which can be easily joined by agents who are aware of the communication rules - but unaware of their mode of implementation (Sycara et al., 2003).

Another approach, which could fall under a 2c category in the categorisation above, is convention based languages (Jones and Parent, 2007). Convention based languages have the advantage of not being based on the members' internal mental states and, hence, are not limited by the characteristics discussed above. However, it is not clear how endogenous signal creation can occur. One cannot ensure that every new convention can be defined using the existing logical framework. For the communicative model to be flexible, new conventions need to be introduced during runtime. As Skyrms says "Rather than focusing exclusively on pure conventionality, we should also bear in mind cases where there are degrees of conventionality associated with degrees of plasticity in signalling" (Skyrms, 2010) (p.31). Convention needs to allow for plasticity, it needs to allow for new signals to be created during runtime.

To summarise, several communication models have been suggested for use in MASs, varying from inflexible models that try to define every single possible interaction, to more sophisticated commitment-based and mentalistic models. Every model has its advantages and drawbacks, and each model could be a good fit for a specific type of systems. As seen, none of these models meets all the criteria defined for open self-organising systems. The main objection against the existing models is that they assume that agents can 'speak' the same language and that they have special mechanisms for understanding and interpreting other agents' internal mental states. As with the notion of role, this thesis will suggest a communication model based on the notion of behaviour, seeing signalling acts as behaviours and the environment as the main means for classifying these behaviours as communication.

## 4.4 Environment and stigmergic communications

This thesis provides a descriptive framework for designing open self-organising systems based on a behaviouristic definition of the notion of role. As a result, given the limitations of the current communication models noted above, it will investigate the alternative of environment mediated communications in order to provide a behaviour-based communications model.

This section discusses the notion of environment in MAS, introduces and dis-

cusses the notion of stigmergic communication and, finally, describes a signalling act in behavioural terms.

#### 4.4.1 Environment in MAS

The system’s environment should not be seen as a passive entity of the system. It should be seen as an essential part of a MAS as it provides the constraints in which the system can operate (Weyns et al., 2007) and plays an active role in the re-organisation of its structure (Di Marzo Serugendo et al., 2006). As Parunak says (Parunak, 1997), the environment is often ignored when building systems of electronic agents. However, in real world applications the environment proves to be ‘embarrassingly active’ to be ignored.

The environment provides the space and the conditions in which the MAS operates and, as a result, constrains the agents. For example, a computer network with low bandwidth would constrain the data transferring and communication capabilities of the agents irrespective of the agents’ abilities. It is also agreed that the environment is a dynamic entity, changing states and triggering changes to the organisational structure of the system (Sycara, 1998). Changes to the environment can trigger changes to the way the system operates, and possibly changes to the system’s structure (Bernon et al., 2006).

Despite the fact that the system’s environment is important, this notion is not well defined (Weyns et al., 2007). Different definitions exist, varying from informal descriptions to well defined structures. For example, it is generally accepted that we can consider all the non-agent elements of a MAS as environment. Parunak (1997) sees the environment as a higher level agent. He defines it as  $\text{Environment} = \{\text{State}, \text{Process}\}$ , where ‘State’ is a set of values that define the environment and ‘Processes’ are internal mechanisms that allow the environment to act, to reach different states. As per this definition, the environment is unbounded, it can change its state whenever it wants based on internal processes. The fact that the definition contains processes and states, like the agents have processes, input and output in Parunak’s definition, means that the environment is seen as a higher level agent.

FIPA defines the environment stating that it “provides the conditions under which an entity (agent or object) exists” (Odell et al., 2003b). The physical environment is what restricts the processes that support the existence of various entities.

Many models see the environment as the infrastructure of the system (e.g. (D’Inverno et al., 2012)). The infrastructure usually has certain responsibilities on the system (Weyns et al., 2007), for example to act as a communication

channel, or to act as a social control mechanism (D’Inverno et al., 2012).

Weyns et al. (Weyns et al., 2007) suggest an interesting definition of the environment. They see the environment as an independent block in the MAS which provides agents with the conditions to exist and also mediates their communication and access to resources. This definition is interesting as:

- It sees the environment as an independent building block of the system, not as a piece of infrastructure or something that only has specific responsibilities.
- It states that the environment needs to be taken into account in every system; it provides the agents with the conditions to exist.
- Since the environment mediates the agents’ communication and access to resources, the environment plays an active role in the system. It does not simply facilitate the communication, it mediates it, meaning that it can intervene and alter it. This idea is similar to Parunak’s {State,Process} definition where the environment has internal processing mechanisms and can alter its state.

Both the definitions provided by (Parunak, 1997) and (Weyns et al., 2007) contain the characteristic that the environment is an active, separate entity with its own states and, potentially, its own behaviour. This idea will be used in the communications model in this thesis. For the purpose of this thesis:

The **environment** provides dynamic conditions for agents to exist and coordinate in an organisational structure.

Where:

- **Provides dynamic conditions to exist.**

This means that without the environment, the agents would not be able to exist. The environment is part of the world where the agents operate and work with each other. The fact that the conditions are dynamic, it means that these conditions might change overtime by the environment itself.

- **Coordinate in an organisational structure.**

The environment provides the means for the agents to communicate, carries stimuli, produces stimuli itself and is altered by agents’ behaviours, aiding the realisation of the notions needed for an organisational struc-



ture to exist; the roles, the communication method and the social order mechanism.

#### 4.4.2 Environment mediated communications

As mentioned above, researchers have investigated the use of environment mediated communications in MASs. Their suggestion is that the environment should not be seen as a passive component of the system, a place where direct communication channels lie on. It should be seen as an active entity used for indirect communications, an entity which is influenced by the agents' behaviour and stores and transmits useful information. The term used to describe this mode of communication and co-ordination is 'stigmergy'. The notion of stigmergy was introduced in the 1950s by the zoologist Grassé to describe the communication methods of ant colonies. Stigmergy is defined by Grassé (1959) as the phenomenon "of indirect communication mediated by modifications of the environment" (also see (Marsh and Onof, 2008), for a review).

The idea of stigmergy is the following: the members' behaviour influences the environment leaving implicit signals, 'stigmata', which can be used for communication. For example, in an ant colony, when an ant is carrying food it leaves a trace of pheromones behind. As a result, when it returns to the nest with food, all the other agents-ants can follow this trace to find the food source. The environment, hence, has been modified from the pheromones and this act communicates to the other ants the path to the food source. The pheromones evaporate during time, so while ants keep using the same path, the trail becomes stronger and more ants are attracted to it. Since ants leave pheromones only when they carry food, when the food source becomes unavailable the pheromones evaporate, the trail becomes weak and stops attracting ants.

Grassé's stigmergic theory and the notion of environment mediated communications have been used not only in biology but also in other sciences such as in sociology and in artificial intelligence. In the design of multi agents systems, stigmergic communications are investigated in order to be used as a means of indirect communication between agents. Instead of using dedicated communication channels, agents' behaviour influences the environment leaving indirect signals to other agents. For example, the fact that an agent is consuming a lot of resources from a network, will communicate the message that more network resources are needed or that this local environment cannot be used for other tasks.

An example of stigmergic communications in MASs is the use of digital pheromones (Weyns et al., 2007). Digital pheromones are structures in the

environment that have the same characteristics as natural pheromones; they can be aggregated if additional pheromones are added, they evaporate overtime, and they diffuse into the local environment. Sauter et al. (2005) for example, have suggested the use of digital pheromones for controlling unmanned vehicles. Their model consists of special ground agents-nodes which are capable of storing different 'flavours' of digital pheromones. When the vehicles go near one of these nodes, they can sense the flavours on the node, make a decision based on them and also deposit (or not) the same pheromone or a different flavour of it.

Another example is the use of distributed computational fields (see for example (Mamei et al., 2004)). Computational fields are inspired by gravitational fields. These models assign properties to the environment, which act as gravitational fields to the agents. If a certain point in space has a greater gravity (or presence in Mamei's model), it will attract more agents. The agents will 'sense' this presence.

Computational fields, like digital pheromones, use the environment by assigning attributes to it, or by altering it in some ways, to inform agents about different situations. It needs to be noted that both these models deal with spatial awareness and coordination of agents, which is only a small part of the communication requirements in a system.

Some researchers have expanded stigmergic theory categorising communication signals into explicit (direct) and implicit (indirect) (Tummolini and Castelfranchi, 2007). They define explicit signals as the communication signals that have been specifically designed, developed or evolved to serve the purpose of communication. For example, speech acts are considered as explicit signals since speech has been developed to allow humans to communicate. Implicit signals are defined as the communication signals that are produced by an agent's behaviour. In the pheromones example above, the fact that ants leave traces of pheromones consists of an explicit signal because this hormone has been evolved to serve communicative purposes. However, the fact that ants follow these traces and, hence, make the path stronger is implicit as the ants' behaviour creates a communication signal; the fact the the path is strong means that the food source is still available.

This thesis examines stigmergy and environment mediated communications in general as it is believed that they offer some interesting characteristics that can be used in the design of the communications model of open self-organising systems.

### 4.4.3 Characteristics of environment mediated communication

Environment mediated communication (also abbreviated as EMC hereafter) is by definition different to conventional means of communication and it has some characteristics that make it an interesting method of communication to be used in the specifications framework provided in the thesis. By conventional means of communication, one means the methods discussed above and in particular methods based on speech acts, the so-called mentalistic methods (Jones and Parent, 2007).

It is evident that speech acts, being explicit in nature, offer a wider range of messages than the implicit messages produced in EMC. However, this thesis argues that environment mediated communications meet the characteristics defined above for the communications model that is needed in an open self-organising MAS.

Firstly, a characteristic of EMC is that information is produced by the agents' behaviour. As mentioned in 4.3, most ACLs take advantage of explicit signals based on speech act theory. This means that agents use explicit, pre-defined signals to communicate with other agents. EMC takes advantage of implicit signals generated by the behaviour of an agent. For example, if a Broker is placing too many buy orders on a stock, because they have a large order to fulfil and not because they want to communicate something to the environment, this behaviour generates a signal, implicit in nature, to other Brokers that are observing the environment. Moreover, most ACLs aim to describe communication as the transmission of information regarding an agent's mental states to other agents. Given that the notion of role has been defined as pairs of stimuli and generated behaviours, this characteristic ties in with the requirement for decoupling; for the communication model being linked to roles instead of individual agents and their abilities.

In addition, since EMC is not using predefined messages, it avoids making any assumptions about the internal structure of participating agents. Information about the environment is exchanged through the environment without the use of a predefined language, or protocol.

Also, the fact that every behaviour of an agent can potentially be a new signal, means that new signals can be created on runtime. There is no restriction for these signals to be reliable, allowing for deceitful communication (Searcy and Nowicki, 2005).

Furthermore, in a system that uses environment mediated communications, the communication channels are not predefined. This means that there are not

single points of failure. As seen above, the KQML model requires an overarching agent to facilitate the communication between two remote agents. In EMS, information is exchanged using the alterations happening to the environment due to the agents' behaviour. This characteristic can make the system more flexible and robust (Sauter et al., 2005) as the communication links are not dependent on specialised agents, communication is decentralised. For example, in ant colonies, an environmental change which leads to a prolonged period of starvation results to a nest's reorganisation. Prolonged starvation leads to changes in behaviour and to changes in the organisation's structure, with more ants becoming the group's foragers (Depickère et al., 2008).

This lack of centrally controlled communication channels allows for the MAS to be flexible and have a greater degree of decentralisation than most of the traditional intention-based models. As the local environment plays a crucial role in communications, agents acquire a good understanding of the interactions performed locally reducing the need for centralised control.

Finally, EMC, and specifically stigmergic systems, have the characteristic of memory. It is observed in lower insect societies, for example in ant-societies, that pheromones evaporate with time. If a path is no longer used by the ants-agents, pheromones will no longer be laid to the ground and, hence, the path will stop being marked after a period of time. On the other hand, if many ants-agents use a specific path, bringing food back to the nest, the pheromone trace will become stronger attracting even more ants. This provides the system not only with the ability to remember actions, as in the case of a frequently used path, but also with the ability to self-adapt and forget irrelevant information, as in the case of an abandoned path. I hypothesise that this characteristic can be really useful in a self-organising decentralised MAS. For example, the co-ordination of agents using a network resource can be designed to work in a similar way as the co-ordination of ants seeking food.

Based on the discussion above on environment mediated communications, a definition of environment mediated communications in this context will be provided and combined with the framework for representing roles introduced in chapter 3 above, in order to arrive at a descriptive framework for specifying open self-organising, heterogeneous, competitive MASs. The characteristics of EMC mentioned above indicate that this type of communication can be used to design a decentralised, self-adapting system.

## 4.5 Communications model

As seen in the section above, many agent communication models exist that are based on strong theoretical principles and have been successfully applied to build functional MASs. However, as discussed, the models that exist in the current literature do not meet all the criteria specified in section 4.2 for a model to be used in the design of open self-organising MASs. This section provides a definition of what a signalling act is in this context as well as a definition of what constitutes successful communication.

This definition is not intention-based or convention-based, it is based on environment mediated communications and on information exchange. As Skyrms (2010) says, “the place to start is not a self-conscious mental theory of meaning, intention, or common knowledge, but rather to focus on information. Signals transmit information, and it is the flow of information that makes all life possible.”.

A **signalling act**, in this context, is:

**Any behaviour that changes the environment and this change carries information about the environment.**

In the equity markets example, the price of a stock rising, the placement of an order, the fact that a Fund has stopped receiving Notices of Execution back from a Broker, all are examples of a signalling act.

It needs to be noted that it is difficult to define when an act ‘carries information about the environment’. Section 4.6 below discusses this in detail based on information theory and on research on the evolution of signals.

If we assume a simple pair of agents, **effective communication** would take place if all the steps below occur:

- The sender alters its behaviour.
- This behaviour changes the environment.
- This change carries information about the environment.
- The receiver understands this change.
- The receiver responds accordingly altering (or not) its behaviour.

It is evident from the steps above that a signalling act alone does not constitute effective communication. The first three steps are enough to say that

a signal has been produced by the sender. This distinction is common in organisations. One's signals might not be received or understood by the intended recipients. Furthermore, the recipients might choose to not respond to specific signals. The response on behalf of the receiver can be to either continue its behaviour or alter it, based on the  $\{S,B\}$  pairs or its role.

#### 4.5.1 Evaluation

The definition of signalling act and the definition of effective communication given above meet the criterion of role-based communication specified in section 4.2 above. It is irrelevant as to what type of agent or which agent performs the signalling act. Moreover, since signals are produced by the behaviour of the role holders, this definition ties with the definition of role given in chapter 3.

In addition, the model suggested does not pre-suppose that the agents have internal mental capabilities. Even if this is the case, it is irrelevant to the model. Agents with mental capabilities are seen as a special case. For example, if the receiver has beliefs about the sender's intentions, this would influence the receiver's behaviour. However, as with the definition of role, this is irrelevant to the definition of the communication model as only behaviours are observable.

Another characteristic needed by the communications model to be used in the design of open self-organising MASs is that it should not rely on dedicated or pre-defined communication channels between the agents. In addition, the model should not be based on a common, pre-defined language or signalling convention. The proposed model meets these two criteria. It does not need pre-defined channels of communication. Signals are transmitted using the agent's local environment, not requiring specific infrastructure to be transmitted. In addition, it is not required that there is a common communication language or signalling convention. Each behaviour that contains information about the environment is seen as a signal. A signal can be an agent's behaviour or a message transmitted in a language. It depends on the receiving agent to process this signal and change her behaviour in order to complete the communication act.

The above also shows that this model allows for the creation of new communication signals. It is a requirement in open systems that not all signals are predefined by the designer. In the model proposed, if an agent's behaviour alters the environment and this contains information about the environment, this constitutes a signal. If this behaviour has not been seen before, because the environmental stimulus is new or the behaviour is new, this is defined as a new signalling act.

Moreover, the proposed model allows for self-organisation and is decentralised. It is independent of roles that control the communication between agents and it does not rely on any special communicative characteristics. Obviously, in the local environment, controlling roles might emerge. For example, an agent might have a behaviour where he re-transmits any information received acting as a central resource of information. This would be beneficial in some particular cases, however it is not a prerequisite of the model.

Finally, the proposed model allows for deceitful communication. The fact that information about the environment is transmitted, does not presuppose that this information is correct (see definition of deceit in section 2.4.1). An example of deceitful communication allowed by the model can be seen by applying it to the blocking practice on the secondary markets.

This practice consists of the following:

1. A Broker has received an order to acquire some stock on a specific price.
2. The Broker reviews the current market status, and places a large order on the opposite side, i.e. to Sell, on a price slightly higher to the one they need to fulfil the order on.
3. They place their original order on the limit price requested.

In the example above, it is not the intention of the broker to execute the Sell order. They have merely placed this order there in order to block the Offers of other brokers and force them to offer to Sell stock in similar prices. This helps the broker to buy the stock, i.e. fulfil the real order, in a good price close to the limit price set by the Fund.

The existence of the large Sell order is a Stimulus, and is an example of deceitful communication. It needs to be noted that this practice is illegal in most of the markets globally, and regulators across the world are trying to impose measures to identify and punish this behaviour.

## 4.6 Informational content

The roles that form the organisational structure are defined based on the behaviour triggered by environmental stimuli. A change in behaviour is invoked when an agent is informed of a change to the environment. When agents communicate, this act needs to be based on the same principle. This is a characteristic that the signalling act described above has: a signalling act needs to contain information about the environment.

As noted above, one needs to define if the information transmitted in a signal is useful, by being relevant to the environment, the quality of information. Traditional information theory deals with the quantity of information in a message, the quality of a communication channel, but it does not talk about the quality of information. The quantity of information is seen as the ratio of the amount of information to be transmitted against the actual amount of information transmitted. To define the communications model in this thesis, a way of measuring the actual quality of information needs to be defined. According to Skyrms (2010), “the informational content of a signal consists in how the signal affects probabilities.”. This definition can be used in the context of designing open, self-organising MASs. Behaviour is invoked by changes in the environment. If the different courses of behaviour of a given role have a certain probability to occur, then a communications signal contains information about the environment if it increases, or decreases, the probabilities of one of these behaviours to occur.

#### 4.6.1 Background

Based on traditional information theory, the quantity of information in a signal about a state is defined as follows:

$$I(s) = \log[\text{psig}(\text{state})/p(\text{state})]$$

Where:

$I(s)$ : Information that the signal contains about the state.

$\text{Psig}(\text{state})$ : The probability of the state given the signal.

$P(\text{state})$ : The unconditional probability of the state.

Dretske (1981) uses this function assuming that  $\text{Psig}(\text{state})$  is always equal to one. This means that when a signal carries the information that an event took place, the event has taken place with a probability of 1. Since a signal can carry information about many states, the average information of a signal can be given by the formula:

$$I_{\text{states}}(\text{sig}) = \sum \text{psig}(\text{state } i) \log[\text{psig}(\text{state } i)/p(\text{state } I)]$$

#### 4.6.2 Current definitions

Dretske is in agreement with Skyrms that existing information theory accounts only for the quantity of information. Based on traditional information theory, he proposes a theory of information that includes the content of information in



a signal. According to Dretske, three points need to be present for a theory of information to account for informational content as well as for information quantity.

In the case of a signal carrying information that a state  $S$  is  $F$ , then it must be the case that:

- The signal carries as much information as would be generated by  $S$  being  $F$ .
- $S$  is  $F$ .
- The quantity of information the signal carries about  $S$  is (or includes) that quantity generated by  $S$  being  $F$ .

Based on the points above, Dretske defines the informational content as follows:

“A signal  $r$  carries the information that  $S$  is  $F$  = the conditional probability of  $S$ ’s being  $F$ , given  $r$  (and  $k$ ), is 1, (but given  $k$  alone, less than 1).”.

This definition is using the logic of traditional information theory to define informational content of a signal. However, two points of critique have been raised against this definition and I believe that two more points need to also be raised.

Firstly, a point of critique to Dretske’s theory by researchers is regarding the objectivity of signals (Fletcher, 2008), (Gjelsvik, 1991). Dretske suggests that information is a commodity, something that exists in nature. As Fletcher (2008) points out, “Dretske defends his point that information is an objective commodity, by arguing that everything else on which information depends is objective.”. However, this contradicts with what he states later on in his work that “whether or not a signal carries information is a question that may not have an objectively correct answer.”.

I believe that the important factor here is the quantity  $k$  used in the definition of informational content. This quantity represents the knowledge that the receiver of the signal has about the local environment. As per Dretske’s definition, the same signal can carry different information depending on the knowledge of the receiver. For example, let’s assume a system with three different states ( $A$ ,  $B$ , and  $C$ ) and two observers ( $R1$  and  $R2$ ). We also assume that only one state will occur at a given time. In the example, the observer  $R1$  knows that state  $C$  cannot be true, while observer  $R2$  does not have that information. The quantity  $k$  is different for each observer. A signal that state  $B$  is not true contains different information according to Dretske’s definition above. From that signal, the observer  $R1$  will have the information that state  $A$

is definitely true while the observer R2 will have the information that it is either state A or C that is true. The same signal contains different information for the two observers as they have different knowledge of the environment. As Dretske puts it, “the repetition of the same piece of information is not information”.

This is contradictory with Dretske’s claim that information is an objective commodity. I believe that Dretske arrives at this result as he links information and knowledge closely. The fact that an observer of a signal has some knowledge about a state, affects the content of the information of that signal. I believe that these two notions should not be so closely interlinked. I agree that prior knowledge of a state allows an observer to use a signal’s information in a different way. However, I believe that the information in a signal is always the same. It is the knowledge which is yielded by that information that might differ.

Another point of critique to Dretske’s definition is that, according to the definition, a signal  $r$  carries exactly the same information that a state  $S$  is  $F$ . It can be argued that one can never be certain that a signal carries the same information as the state that produced this signal. Different conditions might be affecting the channel through which the signal is transmitted. To use Fletcher’s example (Fletcher, 2008), you can never be sure that a pressure gauge carries the correct information about the pressure in a boiler. Due to several factors, such as the distance of the gauge from the boiler or the temperature of the room, the information that an observer receives from the gauge is not the same as the information produced by the boiler.

One could say that an engineer should be able to tackle this by knowing the distance between the boiler and the gauge or the room temperature. Knowing these parameters, he should be able to design a gauge that accurately reflects the pressure in the boiler. These ‘channel conditions’, as Dretske puts it, can be taken into account so that it is certain that the signal contains the same information as the information generated by the source. To make a more accurate design, an engineer might choose to add more channel conditions. For example, he might want to take into account the material from which the pipe is built, or the atmospheric pressure at the area. This list can grow much larger and it is on the engineer to decide which conditions are relevant to the signal and which not.

Dretske tackles this issue by introducing the idea that a signal carries exactly the same information as the state when the existing channel conditions generate no relevant information. The issue here is that what is relevant information and what not is a subjective decision. In addition, you can not be sure about channel conditions that will possibly exist and, of course, you cannot know if they will be

relevant in case they appear. The above means that one can never objectively say if a signal carries the same information as the state. Since information is an objective commodity for Dretske, the above poses an open question about his definition of informational content.

A third point of critique on Dretske's definition is that the second condition that an account of information theory should meet, does not leave room for deceitful messages. According to Dretske, in the case of a signal carrying information that a state  $S$  is  $F$ , then it must be the case that  $S$  is  $F$ . In nature, there are cases where a signal carries the information that  $S$  is  $F$  when the  $S$  is in fact  $G$ . In this case, where  $S$  is  $G$ , the signal would still contain the same information, that  $S$  is  $F$ . I believe that it is irrelevant if the state is  $G$  or  $F$ . The informational content of the signal would be that  $S$  is  $F$ .

To use an example from nature, the Common Toads signal their sizes by the depth of their croaks (Smith and Harper, 2003). A signal of a specific frequency would mean that the originator is of a specific size. However, this is not always the case in nature. A toad of a smaller size could have evolved to have a more deep croak, possibly to attract female toads. In this case, the signal carries the information that  $S$ , the toad, is  $F$ , of a specific size, where the toad is actually  $G$ , of a smaller size. The informational content of the signal is always the same, that  $S$  is  $F$ , while the second point of Dretske's definition does not hold;  $S$  is  $G$ .

Skyrms, on the other hand, believes that any theory of information needs to account for deception. As Skyrms places it (p.73) "any theory that says that deception is impossible is a non-starter" (Skyrms, 2010). In his work he provides a characterisation of deception in behavioural terms. For Skyrms, misinformation exists if a signal increases the probability of a state when this state is not true. This is classified as deception if this situation occurs frequently and benefits the sender of the signal.

Finally, a fourth point of critique, is the third condition that a definition of informational content needs to meet according to Dretske. This third condition states that for a signal carrying the information that  $S$  is  $F$ , "the quantity of information the signal carries about  $S$  is (or includes) that quantity generated by  $S$ 's being  $F$ ". I believe that this condition is the same as the first condition, that the signal needs to carry as much information about the state being  $F$  as would be generated from the state being  $F$ . In the same chapter (p.64), Dretske himself, wanting to separate content from quantity, recognises that the requirement for the signal to 'include' a quantity of information does not refer to purely mathematical comparison of quantities. It is meant to imply something more. However, he does not define what exactly this comparison should mean.

I believe that Dretske refers to the fact that the informational content of the signal needs to contain at least the content generated by the source of the signal.

### 4.6.3 Skyrms' model

Both authors note that, in information theory, information is something different to what we associate with it in every day life. Usually, by the word information, one means the knowledge of an event or the meaning of a signal. As Dretske puts it "information is different to meaning, importance of truth or knowledge". In a signalling system, the information that is included in a signal is only incidentally related to the meaning of this signal. Skyrms follows the same idea; as he notes "the place to start is not with a self-conscious mental theory of meaning, intention or common knowledge, but rather to focus on information." (p.32).

Starting from this realisation both authors develop a theory of information and how this is transmitted in signalling systems. However, they follow different routes after providing their definitions. Skyrms notes that a signalling system does not require any mental capabilities from its members. Based on this note, and using examples from biology, he expands his theory discussing about learning and invention of new signals. Throughout the book Skyrms uses examples of bacteria organisations reiterating that mental capabilities are not needed for a successful signalling system to exist and for information to flow.

Dretske, states that "information is a commodity that given the right recipient is capable of yielding knowledge". However, despite information being a commodity, he associates it with mental ability. Dretske believes that a theory of information without an account of belief is incomplete. This becomes apparent in his statement that "what information is transmitted may depend on what the receiver already knows about the possibilities existing at source.". I believe that this contradicts with the fact that information is just there, as a commodity. Despite the fact that is capable of yielding knowledge, it is contradictory to say that the content of a signal is different depending on the knowledge of the recipient of the signal. The signal might yield different knowledge to different recipients, however the informational content in the signal is always the same.

A considerable part of both books deals with the distinction between informational content and the quantity of information in a signal. As Dretske notes, "information theory tells us how much information is in a signal but it does not tell us what this information is". For example, a signal that communicates the result of tossing a coin contains one bit of information, head or tails. Let's assume that the result is heads. Information theory tells us how much this information is, one bit, and why, it increases the probability of a result from 0.5 to

1. However, it does not provide the means to describe that this signal contains the information that the actual result was heads.

Based on information theory, both authors define the quantity of information contained in a signal about a state as  $I(s) = \log[\text{psig}(\text{state})/p(\text{state})]$ . Where  $\text{psig}(\text{state})$  is the conditional probability of the state to occur given the signal. Probability  $p(\text{state})$  is the unconditional probability of the state to occur at the given time. The logarithm is a base two logarithm. From the formula above, we see that as the unconditional probability of a state increases, the information contained in a signal stating that this state occurred decreases. For example, a signal stating that it rained in an October's day in London contains less information than a signal stating that it rained in a desert. Based on the same reasoning we can say that when the number of possible states in a system increases, and if the states are equiprobable, a signal that carries information about a state eliminating the possibility of other states having occurred, contains more information. For example, let's assume a system with two equiprobable states. When a signal about a state is received, it contains 1 bit of information. However, in a system with four equiprobable states, a signal about a state contains 2 bits of information.

According to Skyrms, traditional information theory tells us how much a signal moves probabilities of a state. In the example of tossing a coin, a signal that says 'Heads' affects the probability of the result being tails to 0. He continues saying that if one needs to define the content of a signal, he would need to define how the signal affects the probabilities of each state. To use the same example, the fact that the signal says 'Heads', changes the probabilities of the state to be heads rather than tail.

The content of information in a signal is the 'how' it affects the probabilities of each state while the quantity of information is the 'how much' it affects these probabilities. As a result, we can define the content of a signal by using a vector which shows how the probability of each state is moved given a signal. For example, in a system with two possible states, the informational content of a signal 'sig' will be given by the following vector:

$$I_{\text{states}}(\text{sig}) = \{\log[\text{psig}(\text{state1})/p(\text{state1})], \log[\text{psig}(\text{state2})/p(\text{state2})]\}$$

This definition provides a simple formula of defining the content in an information signal. However, it is difficult to use it in real life applications because of two drawbacks.

Firstly, this definition presumes that one knows all the potential states of the environment and their probabilities and, hence, is able to define the information vector. In a real life example, there can be potentially many different states. In

large systems, it would be practically impossible to define all these states and also calculate their probabilities at each given time. In a closed system with  $n$  states, the definition of informational content takes the following form:

$$I_{\text{states}}(\text{sig}) = \{\log[\text{psig}(\text{state1})/p(\text{state1})], \log[\text{psig}(\text{state2})/p(\text{state2})], \dots, \log[\text{psig}(\text{staten})/p(\text{staten})]\}$$

The number of states,  $n$ , can be very large, not allowing an observer of the system to know the unconditional probabilities of all the states and their conditional probabilities given a signal. In this case, it would be practically impossible to define the informational content of a signal as it will not be possible to calculate all the components of the vector.

The second drawback of this definition is that it does not cater for open systems. In a closed system, no matter how large it is, one could theoretically define a discrete set of states and their probabilities. In an open system, however, not all states can be defined a priori. This poses a more challenging issue as it is impossible both for the designer of the system and the members of the system themselves to create and calculate the information vectors that Skyrms suggests.

For an open system with, currently,  $n$  states, the definition of the informational content could be given by the following formula:

$$I_{\text{states}}(\text{sig}) = \{\log[\text{psig}(\text{state1})/p(\text{state1})], \log[\text{psig}(\text{state2})/p(\text{state2})], \dots, \log[\text{psig}(\text{staten})/p(\text{staten})], \dots, \log[\text{psig}(\text{statem})/p(\text{statem})]\}$$

Here,  $m$  is a number larger than  $n$ . The states between  $n+1$  and  $m$  represent the states of the environment that **potentially** exist. These are states that are currently unknown to an observer of the system but, the system being open, exist or could potentially exist. It is obvious that one cannot know the probabilities of these states. As a result, it is not possible to calculate the elements of the vector.

In addition, when new states are introduced, the probabilities of all the states change. For example, in a system with  $n$  states, if a new state is introduced, then the unconditional probabilities for all the  $n+1$  states will need to be recalculated. Since the system is open, new states can be introduced at any time. This means that one would need to recalculate the probabilities of all the states if a new state is introduced.

These are the two issues of Skyrms' definition with regards to defining in-

formational content in large open systems. Firstly, it is difficult to know all the states of a system and calculate their probabilities. In addition, the introduction of new states changes the equilibrium of the system. I believe that both these issues could be tackled using the same solution.

#### 4.6.4 Expansion of Skyrms' definition

The suggested solution is the introduction of an ever existing state, Sz which represents non-important, unknown, and not yet present states of the system. This section will first show the use of Sz in a large closed system, then in an open system and based on these two examples, a definition of the state Sz will be provided.

Suppose a large closed system. As discussed above, the informational content of a signal is given by:

$$I_{\text{states}}(\text{sig}) = \{\log[\text{psig}(\text{state1})/p(\text{state1})], \log[\text{psig}(\text{state2})/p(\text{state2})], \dots, \log[\text{psig}(\text{staten})/p(\text{staten})]\}$$

If we make the assumption that at a local level only some states are of importance, then we can reduce the potential states from n to a smaller finite number, let's say StF. This would practically make the unconditional probability of the 'n-StF' states close to zero, but not equal to zero. There will always be a possibility, doesn't matter how small, that a message for these states will be received. This assumption helps us to elevate the unknown game of an undefined number of states into a game with defined states. Based on the introduction of this state, the informational content of a signal would be given by:

$$I_{\text{states}}(\text{sig}) = \{\log[\text{psig}(\text{state1})/p(\text{state1})], \log[\text{psig}(\text{state2})/p(\text{state2})], \dots, \log[\text{psig}(\text{stateStF})/p(\text{stateStF})], \dots, \log[\text{psig}(\text{staten})/p(\text{staten})]\}$$

All the elements of the vector between StF+1 and n, can be combined into the ever existing state described above, Sz, with an unconditional probability close to 0. By doing this, the informational content of a signal would be given by:

$$I_{\text{states}}(\text{sig}) = \{\log[\text{psig}(\text{state1})/p(\text{state1})], \log[\text{psig}(\text{state2})/p(\text{state2})], \dots, \log[\text{psig}(\text{staten})/p(\text{staten})], \log[\text{psig}(\text{statez})/p(\text{statez})]\}$$

This vector gives us the informational content of a signal in a large system tackling the issue of having an uncontrollably large number of states. At this

point, two questions about this vector and the state Sz need to be answered:

1. How does one decide the level of importance of a state.
2. How can one decide where to draw the line between important and non-important states.

Firstly, one needs to define how the states of the environment are split into important (1 to StF) and non-important (Sz). The introduction of state Sz helps us group non-important states into one non-important state. This provides a useful means of reducing the number of elements in the informational vector to a manageable number so that the probabilities of all the important states can be calculated. However, one needs to clarify what it is that makes state StF important and state StF+1 non-important.

This decision mechanism can take various forms and follow different approaches. For example, one can say that the selection of important and non-important states can be done arbitrarily. That an observer has sufficient capabilities of making this decision based on internal criteria. It is obvious that this decision mechanism would not tie with Skyrms' model as it presupposes certain mental capabilities from the part of the receivers.

For a given state, the agents decide on the best action to take based on the expected reward of the action. They act to increase their fitness, as seen in section 3.7. Each action affects the environment, which moves to the next state and, in turn, affects the agent. This new state provides a, positive or negative, reward for the agent. There are several methods for calculating the expected reward from an action; some examples are the finite-horizon reward, the infinite-horizon reward and the average reward.

The finite-horizon method calculates the expected reward based on the formula:

$$\sum_{t=0}^h r_t \quad (4.1)$$

Where:

t is the number of the state.

h is a random threshold.

r<sub>t</sub> is the reward for a given state t.

In a similar manner, the infinite-horizon method is given by the formula:

$$\sum_{t=0}^{\infty} \gamma^t r_t \quad (4.2)$$



where  $\gamma$  is a discount factor for states observed in the future and is a number between 0 and 1.

Here, one needs to note that the reward from a state can be different for each agent. This is based on how much each state of the environment affects an agent. For example, in a small village, a rainy day would have a positive reward for a farmer and a negative reward for a postman. It is important to note, however, that the notion of reward is not a mental notion. The understanding of a reward can be achieved using mental notions, e.g. the postman believes that the rain will make his work harder, but it can also be achieved using behaviouristic descriptions, e.g. the rain actually wets the pavements and this makes the postman's work more difficult.

The theory of reinforcement learning can also be used in deciding which states are of importance and which not. In the same way that reinforcement learning methods are used to decide which action is the most beneficial given a state, it can be used to decide which states are most beneficial, or important, for an observer. Changes in the environment's state cause the observers to behave in a particular way. This behaviour can offer some reward to the observer and/or incur a cost, i.e. a negative reward. The method of calculating the expected reward of an action can be used to calculate the expected reward of a state occurring as follows:

- A state of the environment occurs, a Stimulus is produced.
- An agent selects the most appropriate action for this state. This is the action that provides the largest expected reward.
- Due to this action, the environment changes state, another Stimulus is produced.
- This new state provides a, positive or negative, reward to the agent.
- Based on this reward, the agent decides if the previous state was important or not.

As one can see, this is a process that describes reinforcement learning with the addition of a fifth step where the agent-observer uses the reward of its action to calculate how important the state that occurred was. This calculation can be based on the optimisation methods mentioned above.

For example, following the finite-horizon model, the expected reward for the observer can be given by the formula:

$$E\left(\sum_{t=0}^h r_t\right) \tag{4.3}$$

Where:

$t$  is the number of the state.

$h$  is a random threshold.

$r_t$  is the reward for a given state  $t$ .

Given that the observer needs to optimise its expected reward, only the states that offer the largest absolute reward will be included in the  $h$  states of the formula above. These are states with a very large positive or negative reward. These  $h$  states will be the states that are deemed important at the given time.

The steps above show how an observer can decide how important a state is or not. This is one of the two important steps in the decision mechanism and reinforcement learning gives us a way of making this decision in a behaviouristic manner.

The second step in the decision process is to decide where to draw a line between important and non-important states. The expected reward of a state is a way of sorting states in terms of importance but we also need a way of posing a threshold that defines which states are important and which are non-important and will be grouped into the state  $S_z$ . This is the number ' $h$ ' in the formula above. The original issue here is that the states are too many for an observer to be able to calculate their probabilities at a given state. It is suggested here that this number ' $h$ ' is decided by each observer according to their capabilities. An observer with unlimited capabilities would be able to calculate the probabilities of all the states. However, in a real life example, this is impractical and the observers need to assign a value to this threshold ' $h$ ' based on their capabilities and the outcome of the reinforcement learning process.

This action, the definition of number ' $h$ ' can be part of a reinforcement learning process itself. For example, if an observer selects a very high ' $h$ ', then it will not be able to calculate the probabilities of all the important states before the next state occurs. This would lead the observer to select a lower value for ' $h$ '.

To summarise, the question of selecting which states are important and which are not, can be answered using the existing theory of reinforcement learning. Expanding this theory in calculating, not only the expected reward from an action, but also the expected reward from a state we can assign ranks of importance to the different states. It is a matter of reinforcement learning as well for the agent to decide how many of these states are important and which not. The second point that needs to be addressed is how the informational vector is affected

when a signal for a non-important state is received. The random appearance of new states and signals is a common occurrence in open systems. Addressing this second issue will also help us tackle the issue identified in applying Skyrms' definition to open systems.

In order to the answer this question, one needs to introduce the concept of **local informational content**. Different states can be considered of importance in different places of the environment and by different members. In order to better understand the concept of local informational content, let's assume a closed system with four nodes/observers (A, B, C and D) and three possible states (S1, S2, S3). In this example, when a signal Sig is received at a given time, the informational content for each node is given by the following vectors.

$$IA(\text{sig}) = \{\log[\text{psig}(S1)/p(S1)], \log[\text{psig}(S2)/p(S2)], \log[\text{psig}(S3)/p(S3)], \log[\text{psig}(SzA)/p(SzA)]\}$$

$$IB(\text{sig}) = \{\log[\text{psig}(S1)/p(S1)], \log[\text{psig}(SzB)/p(SzB)]\}$$

$$IC(\text{sig}) = \{\log[\text{psig}(SzC)/p(SzC)]\}$$

$$ID(\text{sig}) = \{\log[\text{psig}(S2)/p(S2)], \log[\text{psig}(SzD)/p(SzD)]\}$$

Note that the elements in each vector are just an example that shows that different elements can be in each node's vector at a given time. It can be argued that all the vectors above contain the same elements with some elements being merged into one, the state Sz. For example, for observer C, all the states are merged into one state, SzC. Knowing that the unconditional probability of the state Sz is close to zero, the above makes it easy for each node to calculate the probabilities of each of the important states.

For the sake of simplicity in calculations let's assume that, for each node at a given time, every state is equiprobable. Also, that the probability of the Z state to occur is close to 0, but not 0. The information vectors will contain the following values (where the unconditional probability of the Z states is shown as 0):

$$IA(\text{sig}) = \{\log[\text{psig}(S1)/0.33], \log[\text{psig}(S2)/0.33], \log[\text{psig}(S3)/0.33], \log[\text{psig}(SzA)/0]\}$$

$$IB(\text{sig}) = \{\log[\text{psig}(S1)/1], \log[\text{psig}(SzB)/0]\}$$

$$IC(\text{sig}) = \{\log[\text{psig}(SzC)/1]\}$$

$$ID(\text{sig}) = \{\log[\text{psig}(S2)/1], \log[\text{psig}(SzD)/0]\}$$

In the example above, let's assume that a signal for state S1 is received. The local informational content for each node will be given by the following vectors.

$$IA(\text{sig}) = \{1.58, 0, 0, 0\}$$

$$IB(\text{sig}) = \{0, \log[\text{psig}(SzB)/p(SzB)]\}$$

$$IC(\text{sig}) = \{\log[\text{psig}(SzC)/p(SzC)]\}$$

$$ID(\text{sig}) = \{0, \log[\text{psig}(SzD)/p(SzD)]\}$$

Since state S1 does not explicitly exist in the vectors of nodes C and D, it

looks like no elements are affected by the signal. The receipt of a signal about a non-existing state seems to have no effect on any elements of the vector for nodes C and B. However, this cannot be true as the information carried by the signal does exist. Therefore, some informational content has been transmitted. It is the states Sz that are affected by this signal. The fact that state S1 does not exist in the informational vectors it means that this state was deemed to be non-important for these nodes at that given time. A learning process needs to be introduced here that makes a newly introduced state important and set it as an element of the vector. In the example above, the informational vectors would be converted to:

$$IA(\text{sig}) = \{1.58, 0, 0, 0\}$$

$$IB(\text{sig}) = \{0, 0\}$$

$$IC(\text{sig}) = \{0, 0\}$$

$$ID(\text{sig}) = \{0, 1, 0\}$$

In a more general form, the vectors will look as follows. These updated information vectors are the outcome of the learning process that introduced state S1.

$$IA(\text{sig}) = \{\log[\text{psig}(S1)/p(S1)], \log[\text{psig}(S2)/p(S2)], \log[\text{psig}(S3)/p(S3)], \log[\text{psig}(SzA)/p(SzA)]\}$$

$$IB(\text{sig}) = \{\log[\text{psig}(S1)/p(S1)], \log[\text{psig}(SzB)/p(SzB)]\}$$

$$IC(\text{sig}) = \{\log[\text{psig}(S1)/p(S1)], \log[\text{psig}(SzC)/p(SzC)]\}$$

$$ID(\text{sig}) = \{\log[\text{psig}(S2)/p(S2)], \log[\text{psig}(S1)/p(S1)], \log[\text{psig}(SzD)/p(SzD)]\}$$

The learning process can also have a fallback loop that combines non important states into Sz. This is not a prerequisite for the model, however it can be proven very useful in large applications, clearing up non-important states and making computation simpler.

The introduction of local information content and the learning process can be used to tackle the second issue of Skyrms' definition, the fact that it does not cater for open systems. The combined state Sz contains all the known states that are deemed non-important, as well as any states that have not been observed or introduced to the system before. As a result, when a signal for a newly introduced state is received by the nodes of the system, a new element can be introduced to their vectors to account for that state.

To summarise, Skyrms' definition of informational content is very interesting as it provides a means of representing the informational content of a signal in a mathematical way. However, it has two limitations. Firstly it does not cater for large systems, as it presupposes that the probabilities of all the states are known. Secondly, it does not allow for new states to be introduced. I believe

that these two issues can be tackled with the introduction of three additions to this definition. Firstly, a combined state that contains both non-important and non yet introduced states. Secondly, the fact that different members of the system can have different, local, informational vectors. Thirdly, a learning process that can alter the elements of the local informational vectors.

## Chapter 5

# Social Order

### 5.1 Introduction

Let's consider the example of a software company where each member has a distinct and well defined role. Sales people sell the software to clients, analysts produce specification documents, developers build the software according to specifications, and testers ensure that it functions as specified. If the roles within the company are well defined, it does not matter which person assumes a role as soon as they are qualified for it. If the requirements of a project change slightly, this can be resolved if some of the members do a slightly different job, if the roles get altered. For example, the analysts might need to assist the customer with using the software. If a new need appears in the company, for example the need to create a new product, a member can assume the role of the product analyst to help with this task. Furthermore, if the sales people sell more copies of the software, it is simply needed to hire a few more analysts, developers and testers to ensure that everything is delivered on time and according to the clients' specifications.

The example of this company is an example of an open system, albeit open in a controlled way, with flexible roles where members join and leave the system, taking on different roles and having the ability to alter the existing, or creating new roles. The principal question of this chapter is what is it that makes the company work as expected? How is it ensured that everyone works for the common purpose? What makes the developers build functional software; what makes the sales people to try and sell more software and what makes the analysts to pay attention to what the clients say?

Similar questions apply in any organisation. Even if roles are well defined,

and each member has assumed a role according to their capabilities, how is it ensured that every member will act according to their role? Furthermore, if a member deviates from their role, how is the outcome of this action controlled to ensure that the organisation still works towards its purpose? In other words, how is social order maintained?

Social order can be described as the characteristic of an organisation to behave as designed, producing expected and desirable results. As seen in the previous chapters, it is important for an open system to have a well-defined organisational structure. This structure provides the agents with roles that define how they need to behave under given circumstances. These roles and the interaction between them lead to a system that knows how to operate to achieve its purpose. However, one cannot be certain that all the members will act according to their defined roles or that the system will continue serving its purpose.

On the software company example, the different employees have defined job descriptions and lines of command. The question that rises is what makes the members of the company act based on their job description and to follow the lines of command. One can say that everyone's manager ensures that people's work is according to their role. However, who ensures that managers do their job correctly? Maybe there is a department within the company that evaluates the behaviour of the employees. For example, a Human Resources (HR) department, which deals with issues the employees have and with employees that do not fulfil their roles. However, why does the HR department do its job correctly? And what happens if all the employees of the department are on leave? Someone else can argue that the organisation works correctly since everyone needs to receive their salary at the end of the month. Not performing your tasks might lead to a salary cut or a fine being paid. Doing a really good job might lead to a bonus or a salary raise. However, who ensures that a member who cares less about money will not do a bad job on purpose? Finally, one can say that, since all members work for the same company, they share the general goal of selling more and better software. As a result, everyone does their best according to their role in order to achieve this common goal. Again, no one can ensure that a member won't change their mind and start acting against the best interest of the company.

The example above shows that except for the question of 'what needs to be done', which is answered by the organisational structure, one also needs to answer the question of who controls the order of the organisation and how order is maintained. The organisation's roles define the constraints in which the

members can act. However, as Artikis et al. (2009) place it, actuality, what is the case, and ideality, what should be the case according to the structure, do not always coincide in an open system.

## 5.2 Social order and control in open MAS

The issues described above are also present in designing open Multi Agent Systems. There are certain characteristics of open systems that make social order difficult to achieve. These characteristics are summarised below. (Artikis et al., 2009).

### 5.2.1 Characteristics of open systems and social order

**1) One cannot know the internal architecture of the system's members.**

Different agents, with unknown internal structures and built by different teams might join or leave the system at any given time. As a result, it is not possible to know how an agent would react to any given environmental state. Agent systems are intelligent systems, in the sense that they can behave differently in different situations.

In a closed system, all of the members are designed according to a predefined specification. This is not to say that there is no variance, or that each member behaves in the same manner. Different agents, who are destined to occupy different roles, will be behaving differently in various situations. However, their behaviours will be predefined and known to an observer of the system or to a central controller. For example, when using a well-documented software, one can always predict the software's next actions based on the specification documents. Of course, it is not trivial to maintain social order in a closed multi agent system. Complex social interactions might still occur, and they might be desirable. Agents might not be built according to the specification, or they might be designed maliciously to not adhere to the specification. However, the agents' internal structure is generally known, the expected behaviours are documented and all the members of the system are known. This makes it easier for a controller, albeit a central system-wide one, or a local one, to identify abnormal behaviour and correct it.

**2) The system's members do not necessarily share a common, global goal.**

As in human societies, each agent can have its own individual goals, which are not necessarily in harmony with the system's purpose. In a closed system, all



agents are designed to work for a common goal. In a production line, for example, agents are designed to make the production of a product more efficient working together. In an open system, where agents can join and leave at will, it is difficult to ensure that all agents share a common goal. It is easier for a malicious agent to join the system and try to achieve goals which are conflicting with the overall system purpose.

Having conflicting goals between agents is not necessarily always against the system's global purpose. It is, in some cases, a desired characteristic of agent systems. For example, in the secondary trading market situation, each broker agent has as its goal to sell at a higher price than the other agents and buy at a lower price. A selling agent's goal, conflicts with a buying agent's goal. However, these antagonistic goals actually make the market run properly, where agents provide liquidity, trade with each other and set a price for the goods traded.

However, in an open system it is much more difficult to ensure that agents with contradicting individual goals are working towards the system's global goal. In the example of a trading market, consider an agent that places a lot of sell orders at a low price. This behaviour is conflicting with the other sellers' behaviour and with what would be expected by this agent. This will have as a result the price of the goods to fall radically, crashing the market. In a regulated, closed market, where all agents are controlled and checked before entering the market, this would be difficult to happen. Or, at least, it would be easier to locate the conflicting behaviour and isolate it. In an open system, it is more difficult to identify who is causing the market to crash and fix the issue.

### **3) The environment in an open system is volatile**

Furthermore, except for the possibility of having inconsistent goals, existing goals of the system might cease to be relevant. Since the environment is active and prone to change, existing legitimate goals might become obsolete, or even might be contradicting with the new status of the environment. Let's consider, in the secondary equity markets example, a fund that is trying to acquire stock of a company. This is the fund's goal and they have dedicated all their resources to it, possibly because they want to acquire more than 5% of the available stock and, hence, enter into discussions of acquiring the company. If we assume that the company goes bankrupt, the goal of acquiring their 5% through the secondary market becomes obsolete. The change in the environment, bankruptcy, caused a goal to be irrelevant. At this point in time, and since the fund has invested all of their resources in acquiring the, now worthless equity, they are forced to get off the market leaving the role of fund empty and affecting the social order of the organisation.

### 5.2.2 Maintaining social order in open systems

Due to these characteristics of open systems, a designer needs to address the three following points that might cause issues with regards to maintaining the system's social order (Castelfranchi, 2000): (a) Emergent behaviour, (b) modelling and monitoring an open system and, (c) reconciling individual with global goals.

Emergent behaviour is the term used to describe the computational power of a decentralised system where the designer does not define which part of the system computes what. In traditional software architecture, the designer is aware of what functions need to occur, how these are invoked and which part of the system performs which function. In an open system, where members with different capabilities can join or leave the system, one cannot predict which member will perform which function. Since in an open system the members' behaviour cannot be predicted a priori, emergent behaviour is a characteristic that the designer needs to address.

Emergent behaviour could be categorised into two different types; new behaviours from individual agents or due to social emergence. As discussed above, in an open system different agents might occupy the same role overtime. Since the internal design of these agents is unknown, each agent might bring new and different behaviours to the role. Moreover, new behaviours might emerge due to complex social interactions between agents. The fact that not all members of the system are known a priori, means that unexpected interactions between agents might occur, leading to new group behaviours.

It needs to be noted here that emergent behaviour is not always seen as an issue, or as a negative characteristic of open systems. It is very difficult, or even impossible, to completely predefine an open system's behaviour. With agents joining and leaving the system on runtime and without a means of predicting each agent's behaviour, it is the system's emergent characteristics that can be of use to the designer. For example, new environmental states might call for new behaviours from the system. This situation would make a closed system halt. It is this emergent behaviour that can keep an open system running. There needs to be, however, a way of controlling and monitoring this emergent behaviour so that it is used for the purposes of the system.

As a result of emerging behaviour, it is not possible to know or predict the system's status at a given time. An observer of the system, for example the designer or an agent itself, cannot have a global view of the system. As stated above, this makes it very difficult to monitor and manage the system. Given this, and the fact that the agents are autonomous, it is difficult to ensure that

the agents follow a common goal; the goal for which the system was initiated.

The points above show the need for the system to have a controlling mechanism, a way of ensuring that the autonomous agents can work together and according to the designer's plans.

Castelfranchi (Castelfranchi, 2000) summarises the different approaches taken by the agent research community for tackling the issues described above as follows.

### **Centralised coordination infrastructure**

The first approach suggested by Castelfranchi is a centralised coordination infrastructure that controls the autonomous agents' behaviour. This is the approach that d'Inverno and colleagues (D'Inverno et al., 2012) follow and that other commonly used models, such as the Moise+ model, follow. In the example of the software company above, this approach would be similar to an auditing department reviewing the company's work and making recommendations.

### **Organisational approach**

Other models suggest an organizational approach where pre-defined roles, plans, agreements and norms control the agents' behaviour. This approach is similar to when the managers of the company are expected to control the processes of the company. The AGR model utilises this approach, with gatekeeping agents having the task of maintaining the system's order.

### **Free market**

Another suggested approach is a social control approach that resembles a 'free market' model where incentives, sanctions and punishments ensure the agents' cooperation. For example, the promise that someone will get a bonus at the end of the year, might improve their efficiency.

This free market approach can be proven efficient, but it needs to be combined with other approaches. Let's consider, for example, a trader in a broker that has the incentive of an end of year bonus based on the profits they generate for the company. Based on this incentive, the trader will try to generate profit for the company with any means, legal or illegal. They might try to manipulate the market in order to get the best price possible. If this behaviour is identified by the market, the company might get fined and the trader banned from trading. This is an example where the free market approach based on incentives

does not work in favour of social order. Another control mechanism would need to exist in conjunction.

### **Global view**

A global view approach of social order is an approach where each agent shares the same view of the environment, of the external world. With this approach, the agents are able to cooperate by knowing the entire global structure and state of affairs. In an open system, where the environment is changing and agents can join and leave at will, it is difficult for each and every agent to maintain a global view of the system. Moreover, even if agents could share the same global view, nothing ensures that malicious agents would not try to take advantage of the system.

### **Invisible hand**

An 'invisible hand' approach where no one can understand or monitor the effects of local behaviour on the global emerging behaviour. In this approach, the system is self-regulated.

As Castelfranchi points out, these approaches do not necessarily exclude each other. For example, the approach of Electronic Institutions is a mixture of a centralised coordination infrastructure and an organisational approach with pre-defined roles, languages and scenes.

Artikis and colleagues propose a four-level specification of the social constraints that exist in an open agent system (Artikis et al., 2009). These are the following:

1. The agents' physical constraints. This first level deals with constraints imposed on agents due to their capabilities. For example, an agent might not be able to use a resource, simply because it is not capable of using it. These constraints are not imposed by any social rules but they do affect social order.
2. Institutionalised power. By institutionalised power, one refers to the characteristic of certain organisations where specific agents are 'empowered' by the organisation to perform specific tasks, e.g. a priest is empowered to wed a couple, the manager is empowered to hire new members, etc... . Jones and Sergot (1996) define institutionalised power using the counts-as connective. The phrase "according to the constraints operative in institution *a*, the performance of some act *A* by agent *i* counts as a means of creating state of affairs *B*" can be given as "*A* counts-as *B*".

3. Permissions, prohibitions and obligations of the agents. The third level of specification is the definition of allowed and prohibited actions. These might be the result of institutionalised power but, in the general case, they are different to institutionalised power. If agents follow these permitted actions, then they are considered as 'social', else if they behave against the permitted actions, they are considered as 'anti-social'.
4. The enforcement policies that deal with agents that do not comply with their obligations, agents that are 'anti-social'. The enforcement policies are rules that define when and how an agent is sanctioned if it does not comply with its level of power, permissions, prohibitions or obligations.

It is important to note that the enforcement measures should be separated from the social order policy. In a similar fashion to human societies, where the legislative and the executive power are separated, the rules of the social order should not be mixed with their enforcement. This is one of the principles of social mechanisms set out by (Minsky and Ungureanu, 2000).

### 5.3 Social control in AOSE methodologies

OperA defines social order in terms of roles, constraints, and interaction rules. These are rules and constraints that are observed in two different levels. Firstly, each agent that joins the organisation (called a role enacting agent - rea) is bound to a social contract with the organisation. A contract describes a specific agreement for a rea, prescribing its behaviour in order to meet the expectations of the organisation, and is based on trust between agents. The second level is the creation of specific facilitation roles, which are enacted by mutually trusted agents and are designed to keep the organisation running Dignum et al. (2002b). OperA's approach is an organisational approach.

Electronic Institutions (EIs) follow a different approach. EIs define a normative environment for heterogeneous agents and it is the normative rules of the environment that define the obligations, permissions, prohibitions, violations and sanctions for agents enacting different roles. EIs are based on the infrastructure they run on to enforce the institutional rules defined Sierra et al. (2004). A similar approach is followed by (D'Inverno et al., 2012).

Dignum also suggests a model based on norms, described in deontic logic (Dignum, 1999). He defines three levels on which the social behaviour of an agent is defined. These are:

1. High level conventional norms, which are divided into interpretation rules

and prima facie rules. Interpretation rules define notions such as 'good' or 'reasonable', while prima facie norms define general rules that should be followed, e.g. 'one should obey the authorities'. I believe that, in the author's work, it is not clear how these notions can be defined for use in building software agents.

2. Contract level conventions, which define directed obligations and authorisations, the rules that break these contracts and what happens if a contract is not fulfilled.
3. Private level norms, which are internal rules of the agents.

Other approaches have also been suggested during the past years for designing social order on open multi agent societies. For example, Singh suggests the use of 'spheres of commitment' between agents (Singh, 1999). Singh suggests that commitments between agents can contain all the normative notions used to describe social order such as obligation and permission.

ADELFE also offers an interesting approach with regards to defining social order and maintaining control. ADELFE advocates a local cooperation-driven social attitude, where agents try to keep their cooperative relationships with their local neighbours. When something does not follow the prescribed rules, it is seen as a Non Cooperative Situation (NCS). NCSs can be of different types; Incomprehension, Ambiguity, Incompetence, Unproductiveness, Concurrency, or Uselessness. When agents identify a NCS, they try to correct it as to maintain the cooperation with their neighbours Bernon et al. (2004).

## 5.4 Social order in framework

As seen above, it is difficult to find the balance between the system's openness and the need for control, the agent's autonomy and the need for a common goal, the use of norms and the need to be able to define unambiguous social rules applied to software agents. Revisiting Castelfranchi's approaches for achieving social control, one can see that a centralised coordinating infrastructure, as the one suggested by d'Inverno and colleagues, restricts the agents' autonomy and also does not allow for the system to be truly open. The organisational approach suggested by Artikis and colleagues, can be successful but it is not yet clear how normative notions can be defined in detail for use in software agent design. Moreover, a 'global view' approach is not feasible for large, open MASs.

I believe that an 'invisible hand' approach, as suggested for open-markets, combined with a clearly defined initial organisational framework could be effec-

tive in open agent organisations. Open agent organisations have similar characteristics to open markets. Agents can be co-operating for the same goal or competing against each other, in the same way that the members of a market can be working together or competing against each other. A bottom-up, decentralised, approach to reorganise structures has already been applied in artificial economies. Artificial economies are agent-based models of economic systems, where the artificial macro economic regularities evolve from the local interaction of the economic agents governed by internal procedures (Bedau, 2003). As a result, market wide regularities, such as price structures, change not due to a centralised reform but due to the agents' local interactions.

There are three characteristics that social control needs to have in the framework:

1. Be part of the model, not an add-on to the model. Social control cannot be imposed from outside, it needs to exist as a result of the agents' interactions.
2. The social control model needs to ensure that the system has a desired behaviour.
3. Social control needs to be defined in a way that is unambiguous and possible to specify along with the roles and communication patterns.

Social order will be defined in the framework based on notions borrowed from biology and political sciences. As seen in the roles chapter above, a role is defined as a set of reactions (pair of environmental stimuli and behaviours). Initially, roles are designed so that, if each role-holder performs the prescribed action when the environment changes, the organisation keeps behaving as expected. However, agents are allowed to bring new behaviours to the system and, through these, roles can get altered or new roles can emerge.

At each given point in time, an agent might act in a different way than expected. For a stimulus  $S1$ , and a pair of expected behaviour  $\{S1, B1\}$ , the agent can perform action  $B2$ . I will call this deviant behaviour.

**Deviant behaviour is a behaviour of the organisation's members that is not expected by the role these members hold.**

It needs to be noted that it is not important for the system to know why this agent behaved differently, what it thought or how this was decided. The important aspect is that it did behave differently. Based on an evolutionary approach, this thesis accepts that an agent will act to increase their fitness in the system. The term 'fitness' is not used in a strict biological context here. In

the framework, an agent trying to increase its fitness is trying to increase its life span, achieve its goals or acquire more resources.

Since each agent tries to increase its fitness, in any environmental change it might act as per the expected behaviour or with a deviant behaviour. One could say that existing models try to minimise the possibility of an agent having a deviant behaviour. For example, in (D’Inverno et al., 2012) model the system’s infrastructure has the role of controlling the agents and ensuring that they act as per the prescribed scenes. However, this reduces the autonomy of the system and does not allow for emerging behaviours. Deviant behaviour is sometimes desirable, as long as it results in a desirable equilibrium. It is this probability that one needs to minimise; the probability of deviant behaviour resulting in a non-desirable equilibrium. Diagram 5.1 below shows the difference between the framework’s approach and other approaches.

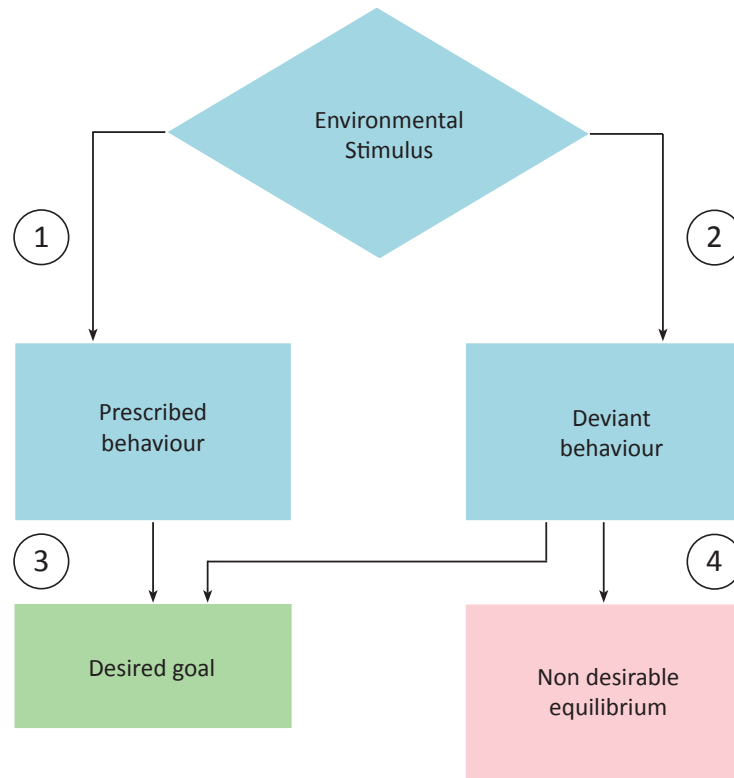


Figure 5.1: Deviant behaviour

Most of the existing social control models try to minimise the possibility of deviant behaviour (arrow number 2) happening. This way, the designer of the system tries to ensure that all agents will act according to their prescribed roles



and, as a result, the system will achieve its desired goal. The framework tries to achieve social order by minimising the probability of arrow number 4; the probability of deviant behaviour leading to a non-desirable equilibrium.

The entire runtime of the system can be seen as a fitness race between the members of the system. Each member is trying to increase its fitness by choosing the appropriate behaviour after an environmental change. The aim of social control in the framework is not to minimise deviant behaviour, but to minimise the probability of this behaviour leading to a non-desirable equilibrium. Here, one needs to define what we mean by non-desirable equilibrium in this context.

**Non-desirable equilibrium occurs when the system reaches a state where it does not serve the purpose it was designed for and this state is irreversible.**

Seeing the system in a holonic view, where each level of the system, from the entire organisation to single roles, can be seen as a black box, one can define the necessary behaviours that constitute the desired equilibrium for each holon. Having this as a basis, and allowing the agents to evolve the roles they play, the system can be flexible, allowing for deviant behaviours but avoiding the state of a non-desired equilibrium. These necessary behaviours are part of the roles of each holon. As Kropotkin (1927) suggests, the answer is a mechanism that identifies ‘wrong attitude’, tries to educate the member that causes the issue or excludes it.

The organisation, itself seen as a holon, tries to impose the necessary behaviours and to ensure that these behaviours exist when the relevant stimuli appear. Drilling down into lower levels, to individual agents, each agent performs the necessary behaviours for maintaining the desired equilibrium. These behaviours produce stimuli, in turn, which can or cannot be linked to other necessary behaviours. This chain of behaviours can be used for identifying situations where the system is leading to a non-desirable equilibrium, what Kropotkin terms as ‘wrong attitude’.

In the secondary market example, with the design of faster trading systems firms have started taking advantage of the technological advancements. This had led to the introduction of certain high frequency traders (HFTs). These firms try to take advantage of higher speed network links and hardware based algorithms. This behaviour, to try to act fast instead of trying to act according to investment decisions, can be seen as deviant behaviour. However, this ‘race for speed’ is not necessarily seen as a negative advancement, as it effectively provides more liquidity to the market and allows for investors to fulfil their orders faster. As a result, it has not been received as a threat, since it leads to

the desired goal; to be able to trade.

Some speculative HFT firms combine fast market data receivers with hardware based algos to enter and withdraw orders quickly, moving the price of stocks in small amounts and entering really small trades when the price reaches a desired point. Other broker firms see this behaviour as leading to a non-desirable equilibrium. These small deviations in price can be considered as a new stimulus, and the resulting behaviour has been to exclude the speculative HFT firms from trading. Other firms have decided to completely leave markets that allow for HFT trading.

As an effect to that, and at a higher holonic view, regulators have decided to try and regulate the HFT firms. This is being done by imposing a tax on HFT trading, on throttling speeds to their networks and implementing tougher market abuse controls.

The example above shows a case where a deviant behaviour, trading faster, is allowed since it leads to the desired goal. However, when HFT traders introduced a new behaviour, to start moving the price quickly and in small tick sizes, a new stimulus appeared; anomalous price movements. As a result, the necessary pair of {Normal price movement, Place an order} ceased to exist, or appeared in lower frequency, and the other members of the system acted since this was leading to a non-desirable equilibrium.

### 5.4.1 Evaluation

I believe that the social order model suggested above meets the characteristics defined in section 5.2.

1. Must be part of the model.

The social order mechanism suggested is part of the organisational model. It is based on defining necessary Pairs for each role, in a holonic view to the different levels of the organisation. Roles are the building block of an organisation and they are defined by Pairs of stimuli and behaviours. Describing the social order mechanism in terms of necessary Pairs for each role, from the high level role of the organisation to the roles of lower level holons, keeps the mechanism as an integral part of the organisation.

2. Needs to ensure that system has the desired behaviour.

Social control is not described in detail in the framework. However, it can be used to maintain social order in an open self-organising system. The framework suggests that a designer should specify necessary Pairs of stimuli and behaviours for each holonic level of the organisation. It is obvious

that one would need to define at least one Pair, at the highest level, which describes the expected behaviour of the organisation. Drilling down into the system's holons, each holon's roles will have necessary Pairs, the roles that potentially form these larger holons will have their necessary Pairs and so on. This creates a network of necessary Stimuli and Behaviours. Since roles communicate with each other, certain roles' behaviours will be producing stimuli that are part of other roles' necessary behaviours. These behaviours might, in turn, be creating stimuli that are necessary for higher level or lower level holons.

As seen in the example of HFT trading above, each role holder, trying to increase their fitness, will be ring-fencing their role's necessary Pairs, excluding members that do not adhere to their roles' necessary behaviours. Having this relationship in different holonic levels, will ensure that this exclusion happens when the system is heading towards a non-desirable equilibrium.

3. Needs to be defined in a programmable way.

The way that social order is maintained in the framework does not assume any internal, or mental, capabilities on behalf of the organisation's members. It is based on a top down view of the system, specifying that each role needs to have certain necessary Pairs. Similar to what was discussed regarding the roles and communication above, the fact that the social control mechanism is defined solely based on behaviours and environmental stimuli, means that there are no restrictions in applying it in a programmatic way.

## Chapter 6

# Framework

The previous chapters described how the building blocks of an organisational structure can be defined using behavioural, non-mentalistic, terms. The notions of role, communication and social order were discussed and described based on the behaviour of the organisation's members. As discussed in the introduction, the aim of this thesis is not to produce a detailed methodology, it is to produce a descriptive framework that can act as the basis of a methodology for designing open self-organising systems using behavioural terms. This chapter provides the descriptive framework for designing such systems.

### 6.1 Complete framework

One can successfully specify an open system as a self-adapting organisation, by specifying the following:

1. The roles that, initially, form this organisation and how these roles can adapt overtime.
2. The communication patterns between these roles and how this communication occurs.
3. The method of preserving social order.

#### 6.1.1 Roles

When designing the organisational model of an open system, one first needs to define the behaviours expected by this system. This refers to, initially, the high level behaviour; what is the system expected to do, its purpose. Through an

iterative process, the designer needs to drill down and specify lower level behaviours, breaking down the high level ones to smaller, more direct behaviours. This approach follows the holonic model Fischer et al. (2003) which sees different levels of abstraction when designing a system. The system can be seen as a black-box, a holon, which has a certain goal, a certain behaviour. This can, then, be broken down into separate holons which have behaviours that, combined together, result in the larger holon's behaviour. In an iterative manner, these holons can be broken down into other holons etc. .

Diagram 6.1 below shows the holonic view of the organisation:

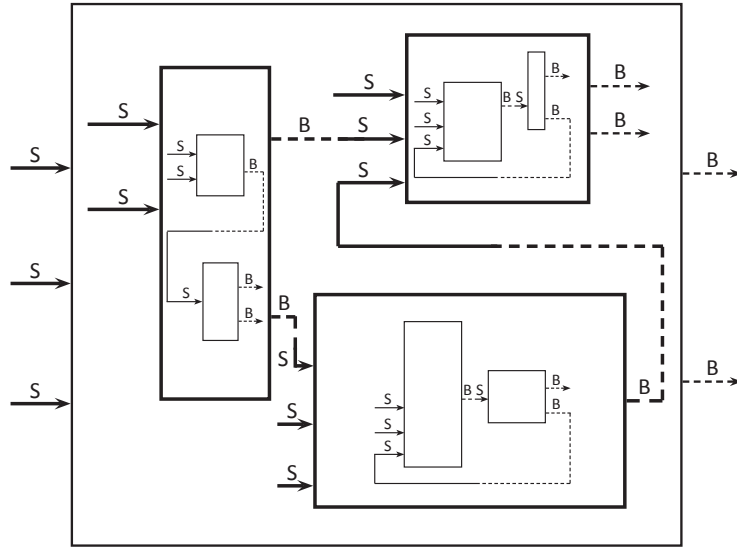


Figure 6.1: Holonic view

The entire system can be seen as a black box (holon) that receives several stimuli from its environment and produces a resulting behaviour. The diagram shows that this holon can be conceptualised as being built by other holons which, in turn, receive certain stimuli and behave accordingly. The iterative process suggested above aims to list the behaviours expected to appear in the different levels of abstraction. The expected behaviours need to be specified in conjunction with environmental stimuli. As discussed above, each behaviour is the result of a stimulus, or of the information that a stimulus has occurred. Hence, behaviours need to be specified in pairs with the stimuli that will trigger them.

As a result of this process, the designer will be presented with a list of stimuli and behaviours that can occur in the system. The next step of the process is to group these pairs based on similar stimuli and behaviours. This is the step

briefly discussed in section 3.5. For economy of resources, and in order to reduce the complexity of the model, the designer needs to identify similar stimuli and group them together. The same process needs to occur in order to group similar behaviours together.

Once the groups of stimuli and behaviours, have been identified they can be placed into pairs of the nature  $P: \{S,B\}$ , where 'P' stands for Pair, 'S' for a Stimulus, and 'B' for Behaviour. The designer needs to decide which pairs can and should belong to the same set so that they can form a role. This step produces the roles that will initially form the organisational structure. There are no explicit requirements as to how this process needs to happen, for example which pairs cannot coexist, or what the limit of pairs per role is. This is an ad-hoc process that needs to be different based on the nature of the system designed. It is expected, however, that these roles will coincide with the holons identified during the iterative process.

### **6.1.2 Communication**

As discussed during the fourth chapter, the members of the system are not stand alone entities. They interact with other members in order to cooperate, coordinate their actions and achieve their goals. In addition, communication is one of the characteristics of the definition of role. Roles fulfil their goals via interaction with other roles. As seen in chapter 4, a signalling act and the notion of effective communication can be defined in behavioural terms. The next step suggested by the framework is to define the signalling acts that are needed by the system to operate and achieve its goals.

The designer of the system needs to define which of the pairs defined above constitute important communication patterns. If more pairs are needed, they need to be introduced to the set of pairs and assigned to different roles. If new pairs are introduced after this step, they need to be reviewed based on the response-threshold model.

### **6.1.3 Social order**

After defining the initial set of roles and the pairs needed for important communications, the designer needs to also specify how the members of the system can continue working towards the global goal, how social order is maintained. As discussed in chapter 5, the goal of the designer should not be to minimise deviant behaviour. Deviant behaviour can be proven useful in several cases where the agents need to take initiatives in order to overcome issues, or in cases

where not every behaviour can be predicted a priori and emergent behaviour is desirable. The goal of the designer is to minimise the possibility of this deviant behaviour leading to a non-desirable equilibrium.

Considering the holonic view of the system, where each level of abstraction can be seen as a black-box performing certain functions, there are certain behaviours that must exist in order for that holon to continue functioning as expected. This is the final point of the suggested framework. The designer needs to identify which behaviours are de facto needed to exist at every level of abstraction. We call these **necessary behaviours** and the pairs that contain these behaviours, **necessary pairs**. An obvious example of a necessary pair, is the pair defined for the holon that represents the entire system. This pair produces a behaviour that is, actually, the goal of the system and needs to be retained during runtime.

In order to provide more flexible systems, an avenue to explore would be setting different thresholds on necessary behaviours. Since a degree of autonomy is required, one could define how much, or for how long, a role holder is allowed to deviate from these necessary behaviours. This way, members of the system would be allowed to slightly alter these behaviours depending on circumstances. This is something that happens in real life organisations. In the software development company example, a sales person could deviate from the behaviour of pricing the software according to the company's schedule if they believe that a deal is very important and needs to be pursued. This point is not covered in the proposed framework, and is noted to be considered in future work.

#### 6.1.4 Framework steps

Based on the above, this thesis suggests that a designer of an Open Multi Agent System for competitive, heterogeneous, potentially deceitful agents, can follow the steps below to reach a good behaviour-based representation of the system's organisational structure. With this structure allowing for self-adaptation of the structure, self-evolution of roles, and environment mediated communications.

##### 1. System

- (a) Describe the business requirement of the actual system.

It is important to describe the purpose of the system, why is it needed. This will help the designer define the scope of the final system and its business function. It will also help to define its limits, its conceivable members, and the environment.

*Outcome:* A detailed description of the scope of the system.

- (b) Describe the function of the actual system (functional requirement).  
This step will describe the high level functional requirement in order to meet the business requirement described above.  
*Outcome:* A high level description of the system's functionality.
- (c) Define Stimuli (Inputs) and Behaviours (Outputs) for the system holon. Group similar S together, and similar B together. Combine Ss and Bs to create pairs P. The final list of Pairs describe the role of the system.  
*Outcome:* A list of stimuli S, a list of behaviours B, and a list of pairs P. These are used to define the purpose of the system in stricter terms.

## 2. Organisational Structure and Roles

- (a) Break down the system holon into other sub-holons. This step is used to produce a preliminary list of roles.  
*Outcome:* A list of sub-holons, and a holonic diagram showing the sub-holons.
- (b) Identify Stimuli (Inputs) and Behaviours (Outputs) for these lower level holons. If possible, the designer needs to group similar Stimuli together, and similar Behaviours together. Combine Ss and Bs to create pairs P.  
*Outcome:* A more detailed list of S, list of B, and list of P.
- (c) Review the Ss, Bs, and group and combine. This is an interim clean-up phase where the designer needs to review the list of Ss and Bs, and group together similar stimuli and similar behaviours.  
*Outcome:* A final list of S, list of B, and list of P for this iteration.
- (d) Repeat steps a to c above, by breaking down the holons into sub-holons. Stop the iterations until roles are granular enough, but not too granular. The designer needs to design the level of abstraction needed for describing the organisational structure of the system.  
*Outcome:* A more detailed list of roles, list of S, list of B, and list of P. More holons in the holonic view diagram.

## 3. Communication

- (a) Review Behaviours from the list created above, and identify any Stimuli created by these Behaviours. This will create some relationships of the form:  $S_x, B, S_y$  where  $S_y$  has possibly been identified as a Stimulus in previous steps. If any new Stimuli are identified, then,



group them, check for relevant Behaviours, and create new Pairs.

*Outcome:* Final list of Pairs.

- (b) Based on the previous step, and the S,B,S relationships, draw the communication lines between the holons-roles.

*Outcome:* An updated holonic diagram with communication lines.

#### 4. Social Order

- (a) Define Necessary Pairs. This is as per the discussion in the Social Order chapter. Pairs that are necessary for maintaining the order of the system need to be defined. The top level Pair has to be Necessary, as it defines the purpose of the system.

*Outcome:* List of necessary pairs.

### 6.1.5 Diagrammatic representation

Figure 6.2 below represents the framework as discussed above:

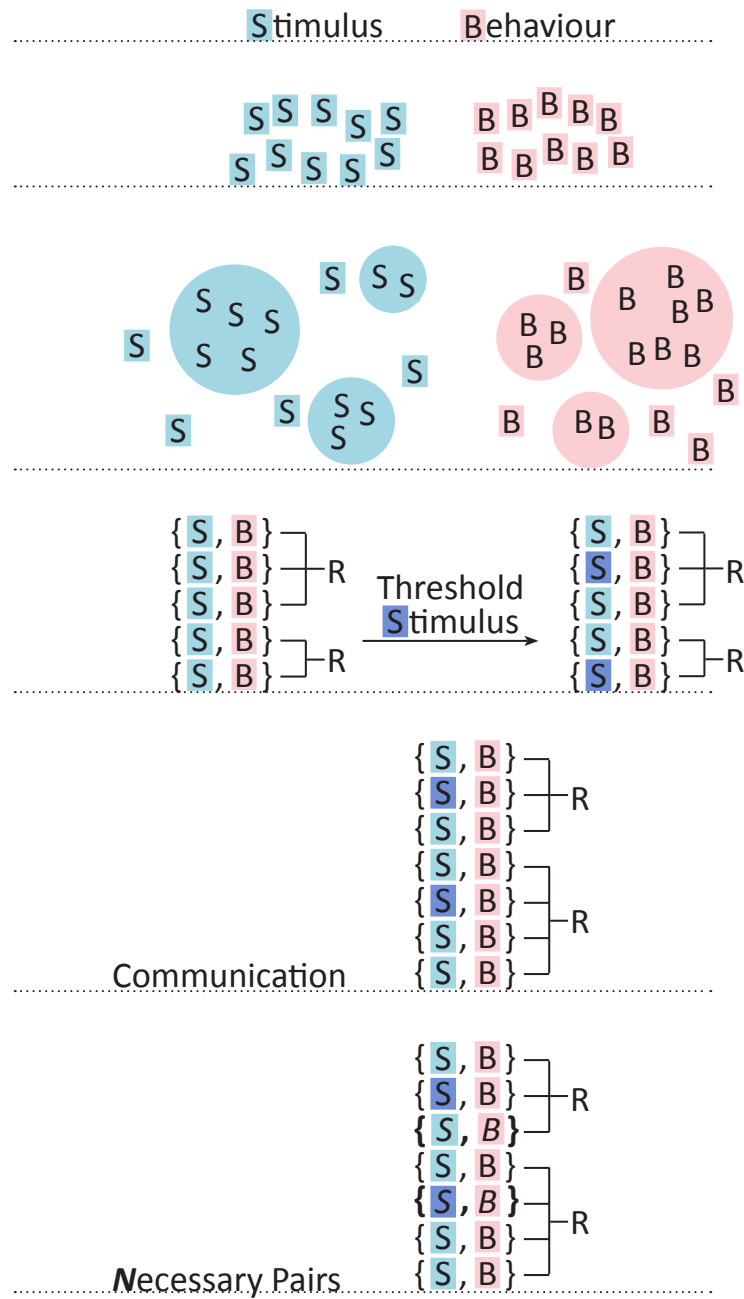


Figure 6.2: Overview of the framework

## 6.2 Use in example domain

This section describes briefly one application of the framework using the example domain, the secondary equity trading markets. As mentioned above, the aim of the thesis is not to provide a detailed methodology. Hence, the example below is not exhaustive, it merely helps for showcasing how the framework can be used in designing open self-organising systems.

### Step 1a, describe the business requirement

Once a company has gone public, by issuing an Initial Public Offering (IPO), its equity titles represent a share in the company, and can also yield annual dividends of the company's profit. Investors seek to trade these equity titles. The places where this trading are the secondary equity markets.

The purpose of the system to be designed is to simulate the behaviour of equity markets. This is needed as markets are an open system, with various participants (heterogeneous), which are competing with each other in order to acquire profit.

### Step 1b, describe the function

The resulting system will simulate the European equity markets landscape where investors are able to buy and sell equity stocks on regulated markets.

### Step 1c, Stimuli and Behaviours for the system holon

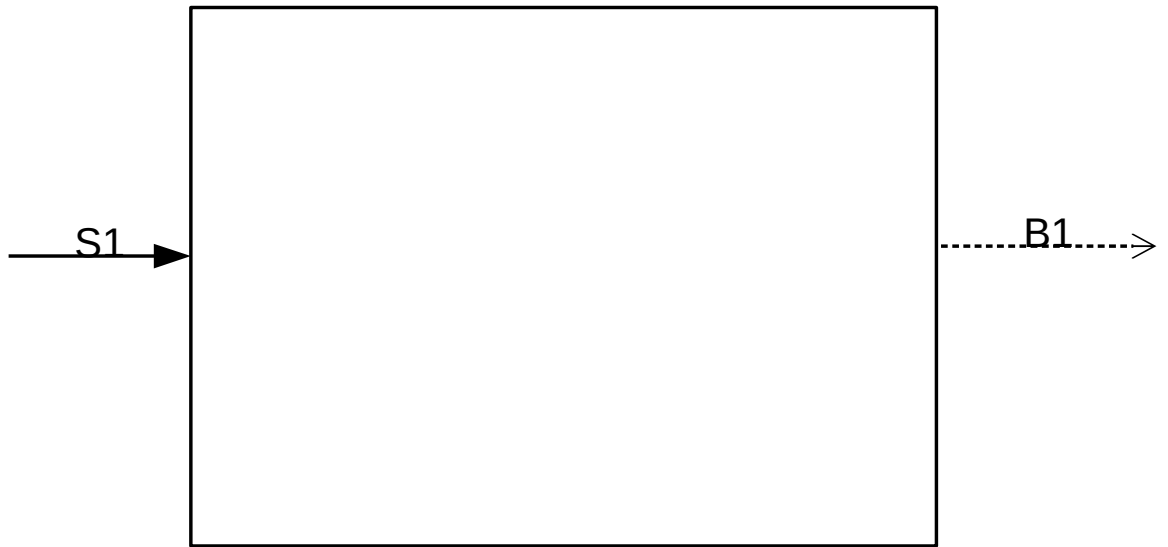
The system holon will have the following Stimuli:

1. S1. Investor invests money

The system holon will have the following Behaviours:

1. B1. Profit is returned.

At this, point we can define one Pair, for the high-level holon: P1 (S1, B1)



#### **Step 2a, Sub-holons**

The following sub-holons are identified:

1. Fund manager
2. Broker
3. Market
4. Clearer

#### **Step 2b, Identify S and B for sub-holons**

The following Stimuli and Behaviours are identified for each sub-holon.

1. Fund manager
  - (a) S2. Receives order to Buy.
  - (b) S3. Receives order to Sell.
  - (c) S4. Calculates it has to Buy.
  - (d) S5. Calculates it has to Sell.
  - (e) S6. Receives instruction to settle portfolio.
- (f) B2. Sends order to Buy.
- (g) B3. Sends order to Sell.
- (h) B4. Settles portfolio.

## 2. Broker

- (a) S7. Receives order to Buy.
- (b) S8. Receives order to Sell.
- (c) S9. Receives instruction that order is traded.
- (d) B5. Sends order to Buy.
- (e) B6. Sends order to Sell.
- (f) B7. Settles trade.

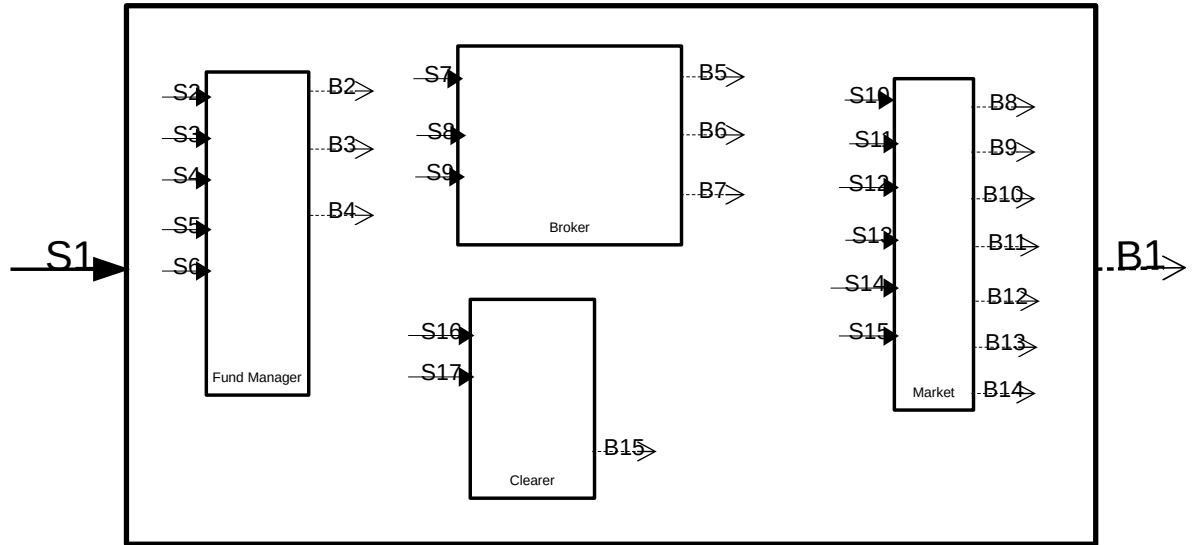
## 3. Market

- (a) S10. Receives order to Buy.
- (b) S11. Receives order to Sell.
- (c) S12. It is time to open.
- (d) S13. It is time to close.
- (e) S14. Price is too far away from limit.
- (f) S15. Volume is larger than the average daily volume (ADV).
- (g) B8. Market is open.
- (h) B9. Market is closed.
- (i) B10. Market is in auction.
- (j) B11. Market is suspended.
- (k) B12. Inform of trade.
- (l) B13. Acknowledges receipt of order.
- (m) B14. Rejects order.

## 4. Clearer

- (a) S16. Receives trade details.
- (b) S17. Receives confirmation of trade.
- (c) B15. Clears trade.

At this first iteration, four distinct roles have been identified (Fund manager, Broker, Clearer, Market) and the initial sets of S and B for each role have been defined. These are represented in the holonic diagram below:



For each role, the following behavioural Pairs have been identified:

- System: P1(S1, B1)
- Fund manager: P2(S2,B2), P3(S3,B3), P4(S4,B2), P5(S5,B3), P6(S6, B4)
- Broker: P7(S7, B5), P8(S8,B6), P9(S8,B7)
- Market: P10(S10,B13), P11(S11, B13), P12(S12,B8), P13(S13,B9), P14(S14,B14), P15(S15, B14)
- Clearer: P16(S16, B15)

### Step 2c, Clean up

At this point, separate Stimuli and Behaviours have been identified by the designer, and assigned to Pairs for each role. This step requires the designer to look for similar Stimuli and Behaviours, and group together so that they are represented as one.

In this example, Stimuli S2, S7, and S10 represent the same stimulus 'Receives a Buy order' and they can be combined into one. The same stands for Stimuli S3, S8, and S11. Doing the same analysis for the Behaviours, B2 and B5 combined. The same for Behaviours B3 and B6.

Grouping the similar stimuli and behaviours together, and re-numbering Ss and Bs, one gets the following:

1. Fund manager

- (a) S2. Receives order to Buy.
- (b) S3. Receives order to Sell.
- (c) S4. Calculates it has to Buy.
- (d) S5. Calculates it has to Sell.
- (e) S6. Receives instruction to settle portfolio.
- (f) B2. Sends order to Buy.
- (g) B3. Sends order to Sell.
- (h) B4. Settles portfolio.

## 2. Broker

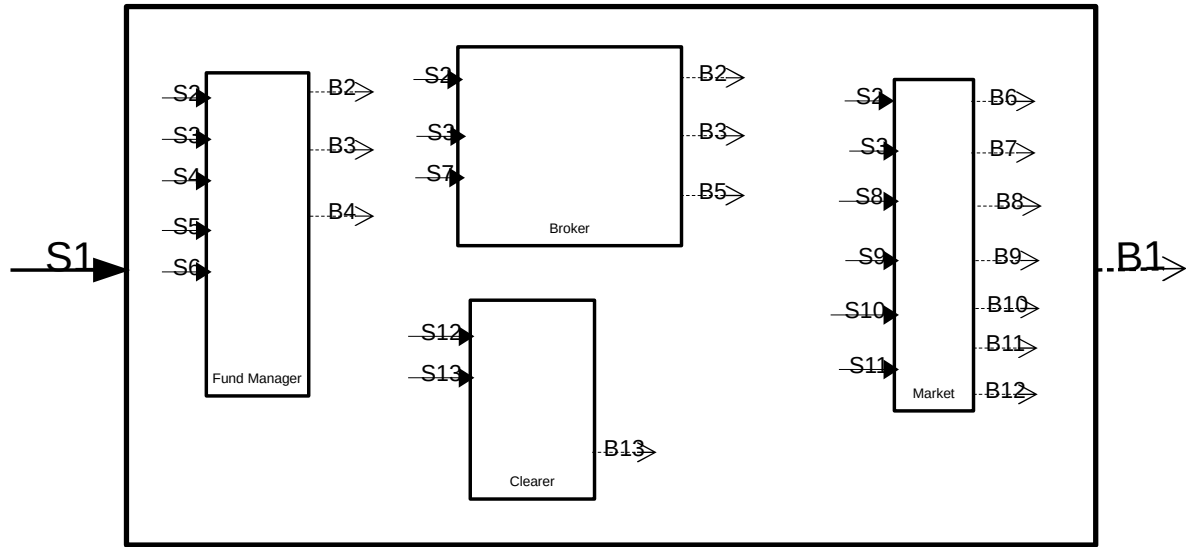
- (a) S2. Receives order to Buy.
- (b) S3. Receives order to Sell.
- (c) S7. Receives instruction that order is traded.
- (d) B2. Sends order to Buy.
- (e) B3. Sends order to Sell.
- (f) B5. Settles trade.

## 3. Market

- (a) S2. Receives order to Buy.
- (b) S3 Receives order to Sell.
- (c) S8. It is time to open.
- (d) S9. It is time to close.
- (e) S10. Price is too far away from limit.
- (f) S11. Volume is larger than the average daily volume (ADV).
- (g) B6. Market is open.
- (h) B7. Market is closed.
- (i) B8. Market is in auction.
- (j) B9. Market is suspended.
- (k) B10. Inform of trade.
- (l) B11. Acknowledges receipt of order.
- (m) B12. Rejects order.

## 4. Clearer

- (a) S12. Receives trade details.
- (b) S13. Receives confirmation of trade.
- (c) B13. Clears trade.



And the following Pairs are identified:

- System: P1(S1, B1)
- Fund manager: P2(S2,B2), P3(S3,B3), P4(S4,B2), P5(S5,B3), P6(S6, B4)
- Broker: P2(S2, B2), P3(S3,B3), P7(S7,B5)
- Market: P8(S2,B11), P9(S3, B11), P10(S8,B6), P11(S9,B7), P12(S10,B12), P13(S11, B12)
- Clearer: P14(S12, B13)

#### Step 2d, review sub-holons and drill down

Now that the basic roles and their Pairs of behaviour have been identified, this steps provides the opportunity to review them and do another iteration of drilling down into the organisational structure by breaking down some holons further. In the example domain, one can say that there are two types of brokers. The Broker role defined refers to an Agency Broker. Agency Brokers do not hold any stock positions at the end of the day, as their purpose is to facilitate Fund managers by offering them access to the market and fulfilling their orders.



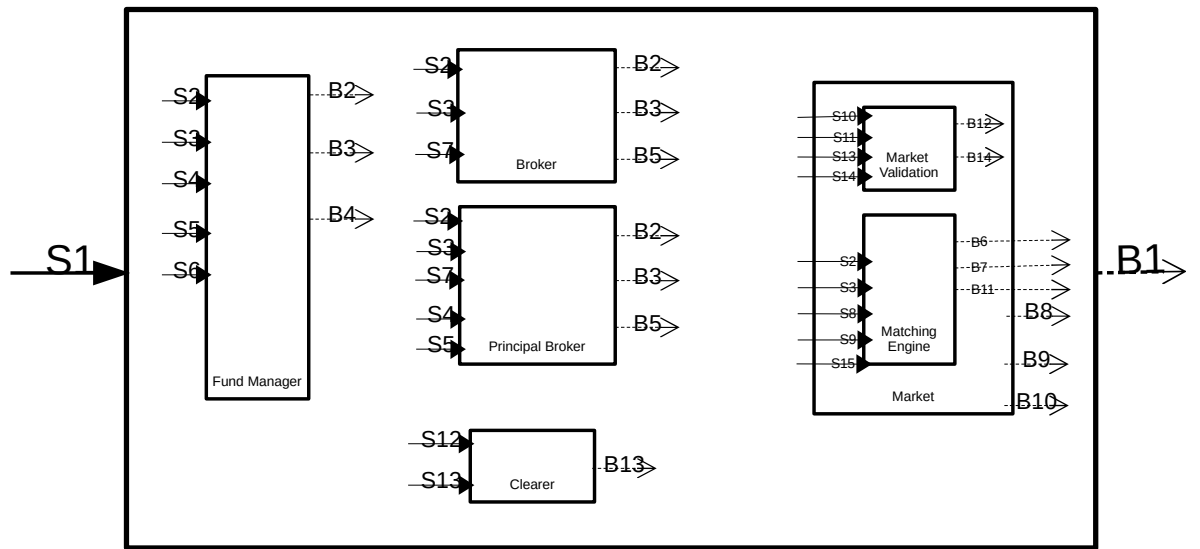
Agency Brokers make profit by charging Fund managers for their services. A different type of Broker, Principal Broker, has also been identified. Principal Brokers offer the same services to Fund Managers, but they are also allowed to hold positions at the end of the day, hence trading for themselves. Principal Brokers make profit based on their trading activity, and also by offering agency services to Fund managers. A new role, of Principal Broker, needs to be defined.

Reviewing the other roles, one can suggest that the Market holon should be broken down into sub-holons, one acting as the matching engine between orders, and one acting as the validation engine which checks the validity of orders according to set rules.

The following roles are defined in this step:

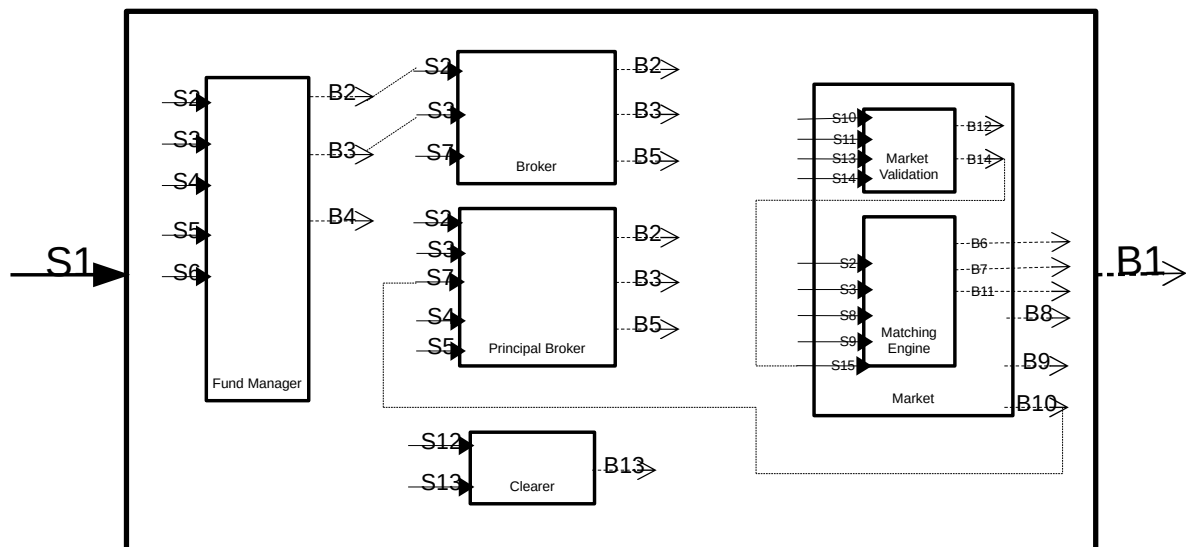
- Principal Broker. The Principal Broker contains the same behavioural Pairs as the Broker (now Agency Broker) role, with the addition of the following two pairs: P4(S4,B2), P5(S5,B3)
- Market Validation. This is a sub-holon of the Market, which contains the following existing pairs: P12(S10,B12), P13(S11, B12). It also contains a new Pair, to communicate with the matching engine: P15(S14,B14) where S14 is the stimulus 'too many rejected orders', and B14 is the behaviour 'Ask to suspend'.
- Matching engine. This sub-holon of the Market contains the rest of the Pairs: P8(S2,B11), P9(S3, B11), P10(S8,B6), P11(S9,B7). It also contains the new Pair, P16(S15, B9), where S15 is the stimulus 'Requested to suspend'.

Note that for the Principal Broker, existing Pairs of the Fund Manager were used. This is the result of identifying the Ss and Bs (as per step 2b), and cleaning up (step 2c).



### Step 3a, Identify $S_x, B, S_y$

The next step requires the designer to identify Behaviours that trigger Stimuli which, in turn, are Stimuli of other Pairs, by reviewing the list of Pairs defined in the previous steps.



#### **Step 4a, Define Necessary Pairs**

The last step of the model is to define the Necessary pairs for the preservation of social order. While making this decision it is important to keep in mind that not all Pairs need to be necessary. Some Pairs, like P1 in the example, have to continue existing for the system to work towards its purpose. While other Pairs would be good to keep existing but are not necessary to preserve the system's social order. For example, Pairs P4 and P5 would be good to exist as they permit the Fund manager to calculate the best course of action over the trading day, allowing more sophisticated fund managers to trade on behalf of their Investors. However, they are not necessary as the system could be based just on the Investors taking this decision using Pairs P2 and P3.

The table below shows all the Pairs identified during this process, to which roles they belong to, and if they are necessary or not.

This example shows that following the steps defined above, a designer is able to produce a list of roles, communication patterns (in terms of  $S_x$ ,  $B$ ,  $S_y$ ), and a list of necessary pairs so as to preserve the social order of the system to simulate the open system of European equity markets.

Table 6.1: Outcome of last step

Pair	Stimulus	Behaviour	Role(s)	Necessary
P1 (S1, B1)	S1. Investor invests money	B1. Profit is returned	System	Y
P2(S2,B2)	S2. Receives order to Buy	B2. Sends order to Buy	Fund manager, Agency Broker, Principal Broker	Y
P3(S3, B3)	S3. Receives order to Sell	B3. Sends order to Sell	Fund manager, Agency Broker, Principal Broker	Y
P4(S4, B2)	S4. Calculates it has to Buy	B2. Sends order to Buy	Fund manager, Principal Broker	
P5(S5, B3)	S5. Calculates it has to Sell	B3. Sends order to Sell	Fund manager, Principal Broker	
P6(S6, B4)	S6. Receives instruction to settle portfolio	B4. Settles portfolio	Fund manager	Y
P7(S7, B5)	S7. Receives instruction that order is traded	B5. Settles trade	Agency Broker, Principal Broker	Y
P8(S2, B11)	S2. Receives order to Buy	B11. Acknowledges receipt of order	Matching engine	
P9(S3, B11)	S3. Receives order to Sell	B11. Acknowledges receipt of order	Matching engine	
P10(S8, B6)	S8. It is time to open	B6. Market is open	Matching engine	Y
P11(S9, B7)	S9. It is time to close	B7. Market is closed	Matching engine	
P12(S10, B12)	S10. Price is too far away from limit	B12. Rejects order	Market validation	Y
P13(S11, B12)	S11. Volume is larger than the average daily volume (ADV)	B12. Rejects order	Market validation	Y
P14(S12, B13)	S12. Receives trade details	B13. Clears trade	Clearer	Y
P15(S14, B14)	S14. Too many rejected orders	B14. Ask to suspend	Market validation	Y
P16(S15, B9)	S15. Requested to suspend	B9. Market is suspended	Matching engine	

## Chapter 7

# Summary

This thesis discussed the issue of designing open, heterogeneous, competitive, self-organising MASs. Open systems where intelligent, autonomous and social members can join and leave at will and self-organising as they can adapt their structure without external intervention.

### 7.1 Questions answered and contribution

The following open questions were set in the introduction of this thesis.

1. The characteristics an organisation needs to have in order to self-adapt its structure. This thesis will examine methods of self-organisation and will argue that a bottom-up, decentralised approach needs to be followed in order to design open MASs.
2. How the notion of role can be formally defined in order to build self-organising MASs. The thesis will compare the so-called behaviouristic and mentalistic approaches in specifying roles in an organisational context and will aim to define the notion of role in strictly behavioural terms.
3. How can environment mediated communications be incorporated into the design of self-organising MASs.
4. Based on the definition of roles in behavioural terms and using environment mediated communications, produce a descriptive framework for designing self-organising open MASs in behavioural terms.

With regards to the above questions:

1. These were discussed in the second chapter where it was argued that open systems need to have a self-organising structure which is described in a non-mentalistic way and can adapt in a bottom-up fashion.
2. In chapter three, several characteristics that a definition of role in open self-organising systems were provided. The most prominent models were examined and compared against these characteristics. Finally, this chapter produced a definition of role that meets the characteristics set.
3. Chapter four discussed speech acts and environment mediated communications (EMC) and argued that EMC is an alternative to traditional agent communication languages. With regards to EMC, it identified the importance of defining the informational content of a signal and expanded Skyrms' definition of informational content.
4. After discussing, in chapter five, how social order can be achieved based on the definition of role provided, a descriptive framework was produced; a methodology for specifying open self-organising systems using behavioural terms. It needs to be noted that the framework is not a detailed methodology, this would be a task too large for the thesis. Further work is needed to produce more detailed steps. However, based on the definitions provided in chapters three to five, it can be said that the framework has the ground and potential to evolve into a detailed formal specifications framework.

This thesis provided the following contributions:

- An expansion to the definition of informational content as provided by Skyrms. This definition can be used to define when a signal contains information about the environment.
- It showed that the building blocks of an organisational structure can be defined based on behavioural terms. It provided a list of characteristics that the definition of role, communication and social order should meet and gave a definition of these notions in behavioural terms.
- These definitions were combined into a descriptive framework which can be the basis of a formal specifications framework for designing open self-organising systems in behavioural terms.

## 7.2 Open questions for future work

The suggested descriptive framework introduces a different way of viewing the design of open self-organising systems, in strictly behavioural terms. However, it

just provides the theoretical background and discussion towards this direction. It is by no means a complete specifications framework that could be used for designing a complete working system. The following points are areas that can and should be considered in the future.

1. Define more commonly used notions in behavioural terms. This would be an extension of section 2.4.1. While the notions used in the discussion for roles, communication and social order are covered in this section, it is expected that more similar notions will appear during the design of a complete specifications framework. These notions will similarly need to be described in a behaviourist way.
2. Explore the possibility of defining a behaviour that groups new stimuli. The grouping of stimuli is not currently defined in a self-organising way. A behaviour which provides this characteristic would need to be introduced.
3. With regards to role re-assignment, introduce the distinction between temporal and morphological polyethism and provide a mechanism for this.
4. An expansion to the model would be to provide the ability to set different thresholds on necessary behaviours. This way, one could be able to define ‘how much’ a behaviour acts as a control of the system’s desirable equilibrium.
5. Produce a worked example using the framework. The level that the secondary market example is used in this thesis is useful for showcasing the framework and discussing its differences with other models. However, an obvious next step would be to use the model to provide a detailed specification and, from this, a worked example.

It is a limitation of a descriptive framework that it is not possible to show that everything will work as expected. The characteristics used for evaluating the definitions of role, communication and social order are accurate and inclusive. However, being descriptive, they are quite generic. Consequently, it cannot be proven that the framework can produce specifications that lead to good, working open self-organising systems. The best way to show this will be through expanding the worked example and implementing a working MAS.

# Bibliography

- H. A. Abbas, S. I. Shaheen, and M.H. Amin. Organization of multi-agent systems: an overview. *Journal of Intelligent Information Systems*, pages 46–57, 2015.
- Y. Abushark, T. Miller, J. Thangarajah, M. Winikoff, and J. Harland. Requirements specification via activity diagrams for agent-based systems. *Autonomous Agents and Multi-Agent Systems*, pages 1–46, 2016.
- J. L. Arcos, M. Esteva, P. Noriega, J. A. Rodríguez-Aguilar, and C. Sierra. An integrated development environment for electronic institutions. In *Software agent-based applications, platforms and development kits*, pages 121–142. Springer, 2005.
- A. Artikis, M. Sergot, and J. Pitt. Specifying norm-governed computational societies. *ACM Transactions on Computational Logic (TOCL)*, 10(1):1, 2009.
- W. R. Ashby. Principles of the self-organizing dynamic system. *The Journal of general psychology*, 37:125–128, 1947.
- R. F. Bales and P. E. Slater. Role differentiation in small decision-making groups. *Family, socialization and interaction process*, pages 259–306, 1955.
- J. Baumann, F. Hohl, N. Radouniklis, K. Rothermel, and M. Strasser. Communication concepts for mobile agent systems. *Mobile Agents*, 1219:123–135, 1997.
- M. A. Bedau. Artificial life: organization, adaptation and complexity from the bottom up. *Trends in cognitive sciences*, 7(11):505–512, 2003.
- C. Bernon, M. P. Gleizes, S. Peyruqueou, and G. Picard. ADELFE: a methodology for adaptive multi-agent systems engineering. In *International Workshop on Engineering Societies in the Agents World*, pages 156–169. Springer, 2002.



- C. Bernon, M. Cossentino, M.P. Gleizes, P. Turci, and F. Zambonelli. A study of some multi-agent meta-models. In *International Workshop on Agent-Oriented Software Engineering*, pages 62–77. Springer, 2004.
- C. Bernon, V. Chevrier, V. Hilaire, and P. Marrow. Applications of Self-Organising Multi-Agent Systems: An Initial Framework for Comparison. *Informatica*, 30:73–82, 2006.
- S. N. Beshers and J. H. Fewell. Models of division of labor in social insects. *Annual review of entomology*, 46(1):413–440, 2001.
- P. J Burke. Role differentiation and the legitimation of task activity. *Sociometry*, pages 404–411, 1968.
- B. Burton. *Role analysis in animals and men*. JSTOR, 1969.
- G. Cabri, L. Ferrari, and L. Leonardi. Enabling mobile agents to dynamically assume roles. In *Proceedings of the 2003 ACM symposium on Applied computing*, pages 56–60. ACM, 2003.
- C. Castelfranchi. Engineering social order. In *Engineering societies in the agents world*, pages 1–18. Springer, 2000.
- M. Cossentino. From requirements to code with the PASSI methodology. *Agent-oriented methodologies*, 3690:79–106, 2005.
- L. R. Coutinho, J. S. Sichman, and O. Boissier. Modelling Organization in MAS: A Comparison of Models. In *Proceedings of the First. Workshop on Software Engineering for Agent-Oriented Systems (SEAS 2005)*, 2005.
- H. K. Dam and M. Winikoff. Towards a next-generation AOSE methodology. *Science of Computer Programming*, 78(6):684–694, 2013.
- M. Dastani, J. Hulstijn, F. Dignum, and J. J. Meyer. Issues in multiagent system development. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 922–929. IEEE Computer Society, 2004.
- M. De Jardin. Multi-Agent Systems: Overview and Research Directions. University Lecture, 2014. UMBC Maryland.
- T. De Wolf and T. Holvoet. Emergence and self-organisation: a statement of similarities and differences. *Engineering Self-Organising Systems*, 3464:1–15, 2004.

- S. A. DeLoach. Engineering organization-based multiagent systems. In *International Workshop on Software Engineering for Large-Scale Multi-agent Systems*, pages 109–125. Springer, 2005.
- S. A. DeLoach and J. C. Garcia-Ojeda. The O-MaSE methodology. In *Handbook on Agent-Oriented Design Processes*, pages 253–285. Springer, 2014.
- Depickère, D. S., Fresneau, and J. L. Deneubourg. Effect of social and environmental factors on ant aggregation: A general response? *Journal of Insect Physiology*, 54:1349–1355, 2008.
- F. Derakhshan, T. Bench-Capon, and P. McBurney. Dynamic assignment of roles, rights and responsibilities in normative multi-agent systems. *Journal of Logic and Computation*, 23(2):355–372, 2013.
- G. Di Marzo Serugendo, M. P. Gleizes, and A. Karageorgos. Self-organization in multi-agent systems. In *The Knowledge Engineering Review*, volume 20, page 165. 2006.
- F. Dignum. Autonomous agents with norms. *Artificial Intelligence and Law*, 7(1):69–79, 1999.
- V. Dignum and F. Dignum. A landscape of agent systems for the real world. *Institute of Information and Computing Sciences, Utrecht University*, 2006.
- V. Dignum and H. Weigand. Toward an organization-oriented design methodology for agent societies. *Intelligent agent software engineering*, 9:191–212, 2003.
- V. Dignum, J. J. Meyer, and H. Weigand. Towards an organizational model for agent societies using contracts. In *AAMAS 02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 694–695, 2002a.
- V. Dignum, J. Vázquez-Salceda, and F. Dignum. OMNI : Introducing Social Structure , Norms and Ontologies into Agent Organizations. *Programming MultiAgent Systems*, 3346:181–198, 2005.
- Vi. Dignum, J.J. Meyer, H. Weigand, and F. Dignum. An organization-oriented model for agent societies. In *Proceedings of International Workshop on Regulated Agent-Based Social Systems: Theories and Applications*, volume 176, 2002b.

- M. D’Inverno, M. Luck, P. Noriega, J. A. Rodriguez-Aguilar, and C. Sierra. Communicating open systems. *Artificial Intelligence*, 186:38–94, 2012.
- F. Dretske. *Knowledge and the Flow of Information*. CSLI Publications, 1981.
- M. Esteva, J. A. Rodriguez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specification of electronic institutions. In *Agent mediated electronic commerce*, pages 126–147. Springer, 2001.
- J. Ferber, O. Gutknecht, and F. Michel. From Agents to Organizations: an Organizational View of Multi-Agent Systems. pages 443–459. Springer, 2003.
- T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an agent communication language. In *Proceedings of the Third international conference on Information and knowledge management*, pages 456–463. ACM, 1994.
- FIPA. Foundation for Intelligent Physical Agents, 2014. URL <http://www.fipa.org/>.
- K. Fischer, M. Schillo, and J. Siekmann. Holonic multiagent systems: A foundation for the organisation of multiagent systems. In *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*, pages 71–80. Springer, 2003.
- L. Fletcher. Information, relevance, and objectivity. *Indiana Undergraduate Journal of Cognitive Science*, 3:3–20, 2008.
- E. Garcia, E. Argente, and A. Giret. A modeling tool for service-oriented open multiagent systems. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 345–360. Springer, 2009.
- E. Garcia, S. Miles, M. Luck, and A. Giret. Evaluating how agent methodologies support the specification of the normative environment through the development process. *Autonomous Agents and Multi-Agent Systems*, 29(6): 1041–1060, 2015.
- L. Gasser. An overview of DAI. *Distributed Artificial Intelligence: Theory and Praxis*, 9(9-29):28, 1992.
- M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge. The belief-desire-intention model of agency. In *Intelligent Agents V: Agents Theories, Architectures, and Languages. Fifth International Workshop, ATAL’98.*, pages 1–10, 1998.
- A. Giddens and P. W. Sutton. *Sociology: introductory readings*. Polity, 2010.

- O. Gjelsvik. Dretske on knowledge and content. *Synthese*, 86(3):425–441, 1991.
- J. J. Gómez-Sanz and J. Pavón. Implementing multi-agent systems organizations with INGENIAS. In *International Workshop on Programming Multi-Agent Systems*, pages 236–251. Springer, 2005.
- P. P. Grassé. La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cu bitermes* sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6:41–80, 1959.
- M. Hannoun, O. Boissier, J. S. Sichman, and C. Sayettat. MOISE: An organizational model for multi-agent systems. In *Advances in Artificial Intelligence*, pages 156–165. Springer, 2000.
- B. Hörling and V. Lesser. A survey of multi-agent organizational paradigms. In *The Knowledge Engineering Review*, volume 19, page 281. 2005.
- J. F. Hübner, J. S. Sichman, and O. Boissier. MOISE+: towards a structural, functional, and deontic model for mas organization. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 501–502. ACM, 2002.
- J. F. Hübner, J. S. Sichman, and Boissier O. S-MOISE+: A Middleware for Developing Organised Multi-agent Systems. In *COIN (Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems)*, pages 64–77. Springer, 2006.
- IBM. Getting Started with SOA, 2007.
- D. Isern, D. Sanchez, and A. Moreno. Organizational structures supported by agent-oriented methodologies. *Journal of Systems and Software*, 84:169–184, 2011.
- M. Jelasity and O. Babaoglu. T-Man: Gossip-based overlay topology management. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3910 LNAI, pages 1–15, 2006.
- A. J. I. Jones and X. Parent. A convention-based approach to agent communication languages. *Group Decision and Negotiation*, 16(2):101–141, 2007.
- A. J. I. Jones and M. Sergot. A formal characterisation of institutionalised power. *Logic Journal of IGPL*, 4(3):427–443, 1996.

- C. M. Jonker and J. Treur. From organisational structure to organisational behaviour formalisation. *International Journal of Agent-Oriented Software Engineering*, 2009.
- J. O. Kephart and D. M. Chess. The vision of autonomic computing. *IEEE Computing*, 36(1):41–50, 2003.
- M. T. Kone, A. Shimazu, and T. Nakajima. The State of the Art in Agent Communication Languages. In *Knowledge and Information Systems*, volume 2, pages 259–284. 2000.
- P. A. Kropotkin. *Anarchism: A collection of revolutionary writings*. Courier Dover Publications, 1927.
- P. Leitão and P. Vrba. Recent developments and future trends of industrial agents. In *Holonic and Multi-Agent Systems for Manufacturing*, pages 15–28. Springer, 2011.
- M. Luck, P. McBurney, and C. Preist. A manifesto for agent technology: Towards next generation computing. *Autonomous Agents and Multi-Agent Systems*, 9(3):203–252, 2004.
- M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent technology: computing as interaction (a roadmap for agent based computing)*. University of Southampton, 2005.
- M. Mamei, F. Zambonelli, and L. Leonardi. Cofields: a physically inspired approach to motion coordination. *Pervasive Computing, IEEE*, 3(2):52–61, 2004.
- L. Marsh and C. Onof. Stigmergic epistemology, stigmergic cognition. *Cognitive Systems Research*, 9:136–149, 2008.
- N. H. Minsky and V. Ungureanu. Law-governed interaction: a coordination and control mechanism for heterogeneous distributed systems. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 9(3):273–305, 2000.
- M. C. Móra, J. G. Lopes, R. M. Viccariz, and H. Coelho. BDI models and systems: Reducing the gap. In *Intelligent Agents V: Agents Theories, Architectures, and Languages*, pages 11–27. Springer, 1999.
- J. P. Müller and K. Fischer. Application impact of multi-agent systems and technologies: a survey. In *Agent-Oriented Software Engineering*, pages 27–53. Springer, 2014.

- M. Nikraz, G. Caire, and P. A. Bahri. A methodology for the analysis and design of multi-agent systems using JADE. *International Journal of Computer Systems Science and Engineering*, 21(2), 2006.
- J. Odell. Agents: A necessary ingredient in todays highly collaborative world. *Business Technology Trends & Impacts, Council Opinion*, 8(1), 2007a.
- J. Odell. Agent technology: What is it and why do we care. *Enterprise Architecture, executive report*, 10(3):1–25, 2007b.
- J. Odell, H. V. D. Parunak, and M. Fleischer. The Role of Roles in Designing Effective Agent Organizations. In *Software Engineering for Large-Scale Multi-Agent Systems*, pages 27–38, 2003a.
- J. Odell, H. V. D. Parunak, M. P. Huget, and R. Levy. FIPA Modelling Area: Environment, Technical Report 030412. Technical report, Foundation for Intelligent Physical Agents, 2003b.
- G. B. O'Reilly and E. Ehlers. Synthesizing stigmergy for multi agent systems. In *Agent Computing and Multi-Agent Systems*, pages 34–45. Springer, 2006.
- L. Padgham and M. Winikoff. Prometheus: A methodology for developing intelligent agents. In *International Workshop on Agent-Oriented Software Engineering*, pages 174–185. Springer, 2002.
- P. Panzarasa, T. J. Norman, and N. R. Jennings. Modelling Sociality in a BDI Framework. In *First Asia-Pacific Conference on Intelligent Agent Technology*, pages 202–206, 1999.
- H. V. D. Parunak. "Go to the ant": Engineering principles from natural multi-agent systems. *Annals of Operations Research*, 75:69–101, 1997.
- H. V. D. Parunak and J. Odell. Representing Social Structures in UML. In *International workshop on agent-oriented software engineering*, pages 1–16. Springer, 2001.
- J. Pavón and J. Gómez-Sanz. Agent oriented software engineering with INGENIAS. In *International Central and Eastern European Conference on Multi-Agent Systems*, pages 394–403. Springer, 2003.
- A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. *KR*, 91:473–484, 1991.

- A. S. Rao and M. P. Georgeff. BDI Agents : From Theory to Practice. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 312–319, 1995.
- G. E. Robinson and R. E. Page. Genetic basis for division of labor in an insect society. *The genetics of social evolution*, pages 61–80, 1989.
- J. A. Rodriguez-Aguilar, Arcos J. L. Sierra, C., M. López-Sánchez, and I. Rodríguez. Towards next generation coordination infrastructures. 2015.
- J. A. Sauter, R. Matthews, H. V. D. Parunak, and S. A. Brueckner. Demonstration of Digital Pheromone Swarming Control of Multiple Unmanned Air Vehicles. 2005.
- M. Schillo, B. Fley, M. Florian, F. Hillebrandt, and D. Hinck. Self-organization in multiagent systems: from agent interaction to agent organization. In *Third International Workshop on Modelling Artificial Societies and Hybrid Organisations (MASHO)*, pages 37–46, 2002.
- H. Schmeck, C. Müller-Schloer, E. Çakar, M. Mnif, and U. Richter. Adaptivity and self-organization in organic computing systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 5(3):10, 2010.
- W. A. Searcy and S. Nowicki. *The evolution of animal communication*. Princeton University Press, 2005.
- J. S. Sichman, V. Dignum, and C. Castelfranchi. Agents’ organizations: a concise overview. *Journal of the Brazilian Computer Society*, 11(1):3–8, 2005.
- C. Sierra, J. A. Rodriguez-Aguilar, P. Noriega, M. Esteva, and J. L. Arcos. Engineering multi-agent systems as electronic institutions. *European Journal for the Informatics Professional*, 4(4):33–39, 2004.
- M. Sims, C.V. Goldman, and V. Lesser. Self-organization through bottom-up coalition formation. In *Proceedings of the Second international joint conference on Autonomous agents and multiagent systems*, pages 867–874, 2003.
- M. P. Singh. An ontology for commitments in multiagent systems. *Artificial intelligence and law*, 7(1):97–113, 1999.
- B. F. Skinner. *Science and human behavior*. Simon and Schuster, 1953.
- B. Skyrms. *Signals: Evolution, learning, and information*. Oxford University Press, 2010.

- J. M. Smith and D. Harper. *Animal signals*. Oxford University Press New York, 2003.
- J. C. Smuts. *Holism and evolution*. Cassell and Company, 1952.
- E. Steegmans, D. Weyns, T. Holvoet, and Y. Berbers. A design process for adaptive behavior of situated agents. In *Agent-Oriented Software Engineering V*, pages 109–125. Springer, 2005.
- J.-P. Steghöfer, H. Seebach, B. Eberhardinger, and W. Reif. PosoMAS: An extensible, modular SE process for Open Self-organising systems. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 1–17. Springer, 2014.
- A. Sturm and O. Shehory. Agent-oriented software engineering: Revisiting the state of the art. In *Agent-Oriented Software Engineering*, pages 13–26. Springer, 2014.
- K. Sycara. Multiagent systems. *AI Magazine*, 19:79–92, 1998.
- K. Sycara, M. Paolucci, M. Van Velsen, and J. Giampapa. The RETSINA MAS Infrastructure. In *Autonomous Agents and Multi-Agent Systems*, volume 7, pages 29–48, 2003.
- L. Tummolini and C. Castelfranchi. Trace Signals: The Meanings of Stigmergy. *Lecture Notes in Artificial Intelligence*, pages 141–156, 2007.
- D. M. Uez and J. F. Hübner. Environments and organizations in multi-agent systems: From modelling to code. In *International Workshop on Engineering Multi-Agent Systems*, pages 181–203. Springer, 2014.
- H. Van Brussel, J. Wyns, Paul. Valckenaers, L. Bongaerts, and P. Peeters. Reference architecture for holonic manufacturing systems: PROSA. *Computers in industry*, 37(3):255–274, 1998.
- D. Van Nieuwenborgh, M. De Vos, S. Heymans, and D. Vermeir. Hierarchical decision making in multi-agent systems using answer set programming. In *Computational Logic in Multi-Agent Systems*, pages 20–40. Springer, 2007.
- J. Vazquez-Salceda, V. Dignum, and F. Dignum. Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 11:307–360, 2005.
- U. Wajid and N. Mehandjiev. A survey of flexible agent interaction approaches. *Multiagent and Grid Systems*, 9(4):247–278, 2013.



- D. Weyns, A. Omicini, and J. Odell. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14:5–30, 2007.
- Whitestein Technologies. Whitestein Technologies, 2016. URL <https://www.whitestein.com/>.
- M. Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- M. Wooldridge and N. R. Jennings. Intelligent agents: theory and practice. In *The Knowledge Engineering Review*, volume 10, page 115. Cambridge University Press, 1995.
- F. Zambonelli, N. R. Jennings, and M. Wooldridge. Organisational abstractions for the analysis and design of multi-agent systems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 235–251. 2001.
- F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 12(3):317–370, 2003.